高技术通讯 2012 年 第22卷 第6期:610~615

doi:10.3772/j.issn.1002-0470.2012.06.009

基于检测驱动的网络蠕虫主动遏制模型①

辛 毅②*** 张庆普* 杨庆海** 金代亮**

(*哈尔滨工业大学管理学院 哈尔滨 150001)

(**哈尔滨工业大学网络与信息中心 哈尔滨 150001)

摘 要 在分析了现有网络蠕虫遏制技术的不足的基础上,提出了一种基于检测驱动的网络蠕虫主动遏制模型:利用网络蠕虫检测技术,在网络出口设置探头对进出网络的数据包进行捕包检测,如发现网络蠕虫的感染主机,则利用网络的漏洞对网络蠕虫进行清除并且修补漏洞;同时采用漏洞扫描的方式对网络内的主机进行扫描,如发现存在漏洞的主机,对漏洞进行修补或对主机进行免疫,以达到防止网络蠕虫感染的目的。给出了这种主动遏制策略的数学模型,从理论上分析了主动遏制过程中各种参数之间的作用关系,为之后的具体策略研究提供了理论基础。

关键词 网络蠕虫,漏洞,检测驱动,主动遏制

0 引 言

Internet本质上是一个开放的复杂巨系统,其结构复杂,缺乏中心控制能力以及其开放性的特征导致存在大量网管层面上的不可控节点,这使得其受网络蠕虫的危害难于控制。这些不可控节点往往缺乏相应的安全防护措施或长期无人管理,一旦感染蠕虫,蠕虫就会长期滞留在被感染节点中,并作为攻击源对Internet始终构成威胁。因此,如何管理、维护那些无序、不可控的网络节点是控制恶性蠕虫扩散、有效降低蠕虫疫情的关键。当前亟需建立一种对抗蠕虫的新体系,以克服传统对抗技术的弱点,提高对新蠕虫的响应速度,及时遏制传染源的扩散,清除网络上那些不可控节点中的蠕虫,堵塞传播蠕虫的漏洞,达到彻底清除蠕虫的目标。

国内外的抗蠕虫研究一般都是从网络的协议入手,研究如何从协议实现上限制蠕虫的快速传播[1,2],这种途径不能从根本上解决蠕虫感染节点的扩散的问题,只能是缓解蠕虫的传播,无法做到对蠕虫的彻底消除。采用传统的方法如防病毒软件,如不能进行及时的升级则不能及时、有效地对蠕虫进行清除和防护。基于良性网络蠕虫的遏制方法的研究目前也仅停留在学术上,值得注意的是如果良性蠕虫不可控[3],对网络及主机造成的安全危害可能会远远超过恶性网络蠕虫,Welchia就是这样的一个例子。网络蠕虫是利用信息系统中存在的漏洞(弱点)实现对系统的攻击的,蠕虫的种类具有多样化的特征,但所利用的漏洞是有限的。主动遏制技术主要是基于对漏洞的遏制,一方面,利用蠕虫检测技术,在网络的出口设置探头进行捕包检测,发现蠕虫的感染主机,利用漏洞进入感染目标主机,对蠕虫进行查杀、对漏洞进行修补或对主机进行免疫等措施达到清除网络蠕虫的目的;另外一方面,采用快速扫描技术对网络内的主机进行扫描,对存在漏洞的进行远程修补。本文针对现有蠕虫遏制技术存在的问题,提出了一种基于检测驱动的网络蠕虫遏制模型,最后给出了对抗策略的数学模型,并从理论上分析了主动对抗过程中各种参数之间的作用关系,为网络蠕虫遏制研究提供了理论基础。

1 漏洞和蠕虫的关系

攻击者或者恶意用户等由于某种原因想要获取信息系统的某些操作权限,以期望能滥用或者损害信息资产。这种威胁增加了信息资产的危险,而其

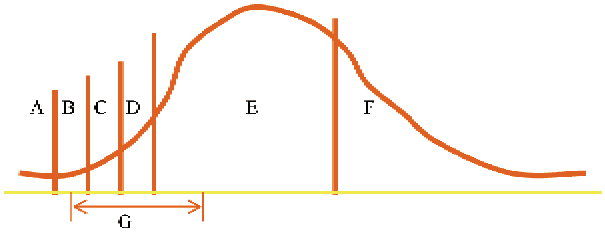
① 国家自然科学基金(61100189)和山东省中青年科学家奖励基金(BS2011DX001)资助项目。

② 男,1973年生,博士;研究方向:网络安全,恶意代码;联系人,E-mail:xinyi@hit.edu.cn

(收稿日期:2011-05-17)

具体行为是通过对信息系统中的漏洞进行发掘利用来执行的。漏洞也就称为攻击者或恶意用户实现其目标的基本的和必需的途径,也是蠕虫得以传播的必要条件。

漏洞从被发现到被广泛使用,再到因受影响的系统的逐渐更新升级而淘汰,存在一个时间周期[4]。漏洞从被首次发现到公开、攻击方法或程序和脚本发布、补丁发布被大量使用直到被淘汰的整个过程称为一个发掘周期,如图 1 所示。



A—高级攻击者发现系统漏洞;　B—非常粗糙的攻击工具被发布;
C—无经验的攻击者使用粗糙工具;　D—自动的扫描/攻击工具被设计;
E—对自动扫描/攻击工具的广泛使用;　F—攻击者开始使用新的漏洞;
G—安全组织发布安全公告。

图 1　漏洞的发掘周期

蠕虫的产生与消亡与漏洞的关系极大,蠕虫生存周期基本上取决于漏洞的发掘周期。既然网络蠕虫可以利用漏洞进行传播,那么我们也可以利用漏洞对蠕虫进行遏制。近几年,一般从漏洞被发现到攻击脚本产生,也就是从 B 到 C,是蠕虫产生的高峰期。利用系统漏洞有效遏制蠕虫传播的方法主要包括以下几个方面:(1)在 A 阶段主动发现系统漏洞,提交到官方软件发行者从而发布官方补丁,在早期预防蠕虫的产生;(2)在 B－D 阶段对发布的攻击脚本进行分析,争取在蠕虫产生前发现网络内的易感主机并采用临时补丁进行修补,防止被新的蠕虫感染,同时对已经产生感染的新的蠕虫进行远程的查杀并对弱点进行修补;(3)在 G 阶段对系统内的易感主机采用官方补丁进行修补,防止蠕虫的感染;(4)在 E－F 阶段对新的蠕虫的产生进行预警和遏制。

2　基于检测驱动的网络蠕虫主动遏制模型

蠕虫的传播适合采用传染病传播模型来描述[5],主要包括简单传染病模型[6]、Kermack-Mckendrick 模型[7,8]、双因素模型[9]、离散解析传播模型[10]及动态隔离和免疫的传染病模型[11]。目前,基于主动遏制的研究比较多的是基于良性蠕虫的对抗方式,而良性蠕虫的引入会带来很多问题。良性蠕虫的可控性一直是困扰这种方式应用的难题,历史上的良性蠕虫造成的危害往往大于其对抗的恶性蠕虫本身[3]。

网络蠕虫造成大规模感染的条件是:接入到网络的计算机存在可被网络蠕虫利用的漏洞。网络蠕虫进入系统后,对可利用的漏洞会采取两种不同的方式:(1)关闭其所利用的漏洞,我们称采用这种方式的为强势网络蠕虫。(2)不关闭其所利用的漏洞,我们称采用这种方式的为非强势网络蠕虫。非强势网络蠕虫可以利用其原有的漏洞进入感染主机。我们发现,一般被网络蠕虫感染的计算机都存在着安全管理问题(如不及时升级,没有安装相应的防护软件或者存在弱口令等严重的安全问题),有可能存在其他的多种漏洞。另外,我们对大量的网络蠕虫样本进行分析发现,网络蠕虫的程序本身一般都存在一些安全漏洞,或者有的网络蠕虫会带有后门(如 Dvldr32 口令蠕虫开放了 VNC Service 远程管理,其密码就为 Stricted)。因此,我们可以充分利用系统及蠕虫程序本身存在的多种漏洞对网络蠕虫进行主动对抗。针对易感主机和感染主机存在大量系统漏洞的实际情况,我们提出了一种全新的基于检测驱动的网络蠕虫主动遏制系统(active worm countermeasure system based on detection,AWCSD)。

AWCSD 模型如图 2 所示。检测与遏制服务器主要完成两项工作:一方面通过在网络出口对进出的流量进行捕获并检测,如在网络内发现感染节点,则利用漏洞远程对其进行清除并免疫从而不被继续感染;另一方面对网络内的节点进行扫描,如发现易感节点则对其进行修补,防止被网络蠕虫感染,在扫描过程中采取自适应的算法,扫描速度根据蠕虫的感染节点的增加而增加。

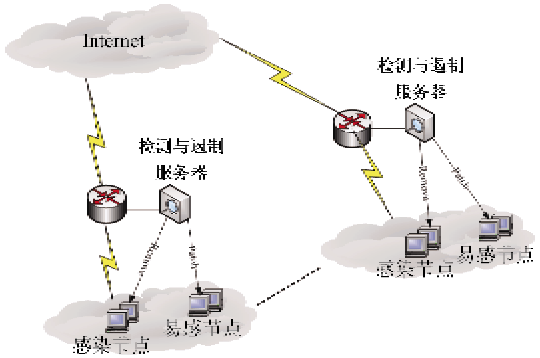


图 2　基于检测驱动的网络蠕虫主动遏制模型

在基于检测驱动的网络蠕虫主动遏制模型中，检测与遏制服务器位于网络出口，通过流量侦听可以检测到网络内的感染节点，利用检测的结果对网络蠕虫进行远程清除；同时利用已知的网络拓扑进行漏洞快速扫描，发现易感主机则对其进行远程修补。与双因素模型相似，网络中的主机的变化可以用图 3 表示。

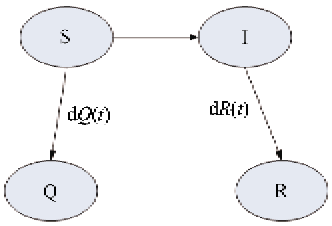


图 3 蠕虫感染状态图

在基于检测驱动的网络蠕虫主动遏制模型的传播模型中用到的符号如表 1 所示。在用双因素模型时，感染主机的清除率和易感主机的免疫率都是常数，这不能反映引入遏制系统后的真实情况，因此在基于检测驱动的网络蠕虫主动遏制的数学模型中我们引入了如下措施：(1)服务器通过侦听全网流量发现感染节点进行远程的清除和免疫(简称远程清除免疫)；(2)服务器自适应地对全网易感节点进行扫描，发现漏洞进行远程主动修补(简称远程修补)。我们将以上两种措施称为对抗工具。由于网络蠕虫的自传播特性，被感染的主机数量以及网络蠕虫的扫描包会按指数规律迅速增长[12]。

表 1 传播模型中用到的符号

符号	意义
N	网络中漏洞主机数量
$I(t)$	t 时刻网络中蠕虫的数量
$S(t)$	t 时刻网络中易感染主机的数量
$R(t)$	t 时刻网络中已感染主机被清除并免疫的数量
$Q(t)$	t 时刻网络中易感染主机被免疫的数量
$P(t)$	T 时刻被远程清除蠕虫和漏洞修补的主机数量
$J(t)$	T 时刻网络中被蠕虫感染过的主机数量 $J(t) = I(t) + R(t)$
β_0	初始时刻蠕虫扫描网络内节点的频率
$\beta(t)$	t 时刻蠕虫扫描网络内主机的频率
γ	感染节点的人工清除率
μ	网络中主机的人工免疫率
η	为根据蠕虫数量 $I(t)$ 调整感染率 $\beta(t)$ 的灵敏度常量
ϕ	单台服务器单位时间内的修复率
τ	调节扫描率对于 $I(t)$ 的灵敏度常量
M	检测与遏制服务器台数

检测系统发现感染节点也会随着增长。为了在网络蠕虫爆发时有效地进行对抗，模型会根据其策略加大扫描的速度，它和感染情况相关：系统发现的网络蠕虫的感染节点越多扫描的速度就会越大，我们称其为自适应地调节扫描。此外，由于遏制系统是由网络管理人员在网络内进行部署的，所以在扫描过程中预先知道网络的拓扑以及 IP 分布，其扫描列表准确可靠，所以在漏洞修补的过程中不会产生对无效地址的扫描，属于 Hitlist 型扫描，因此远程主动修补的速度会很快。由以上分析可知，新引入的函数和 $I(t)$ 以及漏洞检测与修补服务器台数相关，可以表示为

$$\begin{cases} \mathrm{d}R(t)/\mathrm{d}t = \gamma I(t) + \phi m I(t) \\ \mathrm{d}Q(t)/\mathrm{d}t = \mu S(t) J(t) + \alpha(t) S(t) \\ \mathrm{d}P(t)/\mathrm{d}t = \alpha(t) S(t) + \phi m I(t) \\ \alpha(t) = f(I(t)) m = \sigma I(t)^{\tau} m \end{cases} \tag{1}$$

其中 ϕ 为单台服务器单位时间内的修复率，$f(I(t))$ 为单台服务器单位时间内的自适应扫描率，它是关于感染机器数 $I(t)$ 的函数，可由程序算法设定，此处用指数函数表示。$f(I(t)) = \sigma I(t)^{\tau}$，其中 τ 为调节扫描率对于 $I(t)$ 的敏感程度，设 $\tau = 2$。

由于主动修补补丁和远程网络蠕虫清除工具都会给网络带来流量负荷。因此，$\mathrm{d}P(t)/\mathrm{d}t$ 为单位时间内遏制系统修补易感节点和清除的感染节点的个数，并且由于系统硬件、网络带宽和算法效率的限制，$\mathrm{d}P(t)/\mathrm{d}t$ 存在一个上限。故有

$$\begin{cases} \sigma I(t)^{\tau} S(t) + \phi I(t) \leqslant h \\ s_1 \sigma I(t)^{\tau} S(t) + s_2 \phi I(t) \leqslant g \end{cases} \tag{2}$$
$$I(t) \in [0, N], S(t) \in [0, N]$$

其中 s_1 为主动修补补丁的大小，s_2 为远程网络蠕虫清除工具的大小，h 为单位时间内单台服务器处理机器台数上限，g 为单位时间内单台服务器的网络流量的上限，根据实际情况，通过对以上不等式可选择一组合适的参数如下：$\sigma = 1/N^2$，$\phi = 1/N$。

可以得出基于检测驱动的网络蠕虫主动遏制的数学模型：

$$\begin{cases} dS(t)/dt = -\beta(t)S(t)I(t) - dQ(t)/dt \\ dI(t)/dt = \beta(t)S(t)I(t) - dR(t)/dt \\ dR(t)/dt = \gamma I(t) + \varphi mI(t) \\ dQ(t)/dt = \mu S(t)J(t) + \alpha(t)S(t) \\ dP(t)/dt = \alpha(t)S(t) + \varphi mI(t) \\ \alpha(t) = f(I(t))m = \sigma I(t)^{\tau}m \\ \beta(t) = \beta_0[1 - I(t)/N]^{\eta} \\ N = S(t) + I(t) + R(t) + Q(t) \\ I(0) = I_0 << N;\ S(0) = N - I_0; \\ R(0) = Q(0) = P(0) = 0 \end{cases} \tag{3}$$

3　仿真实验与分析

3.1　不同模型之间的对比分析

为了进一步揭示 AWCSD 模型的传播规律和效果,我们通过 Matlab 软件仿真验证了数学模型。设网络中全部主机的数目 $N = 1000000$, 初始感染主机 $I_0 = 1$, $\eta = 3$, $\gamma_0 = 0.05$, $\mu_0 = 0.06/N$, $\tau = 2$, $\beta_0 = 0.8/N$, 检测与遏制服务器台数 $m = 100$ 时,我们将双因素模型、经典疫情模型(Kermack-Mckendrick)和 AWCSD 模型的曲线分别在图中进行比较,得出了相应的主机数量变化曲线。

图 4 表示正在感染的节点函数 $I(t)$ 的变化过程,图 5 表示 $J(t) = I(t) + R(t)$ 及感染过的节点的变化过程,图 6 表示 $Q(t)$ 的变化过程。从图中可以明显看出,引入了感染节点的远程清除免疫和远程修补函数后,网络中感染主机清除免疫和易感主机的漏洞修补数大大增加,同时 $I(t)$、$J(t)$ 大幅降低,和其他两种模型相比,AWCSD 模型能够很快地抑制网络蠕虫的传播。

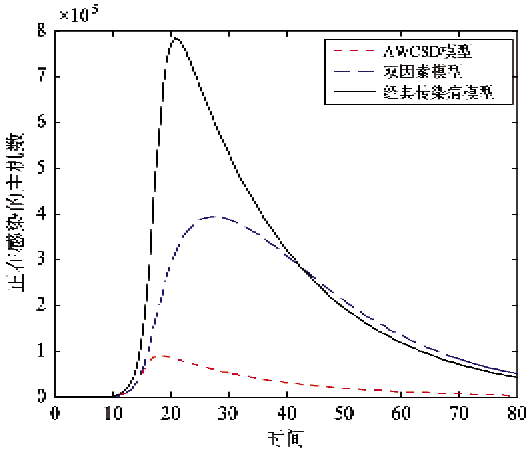


图 4　不同模型正在感染节点的变化对比

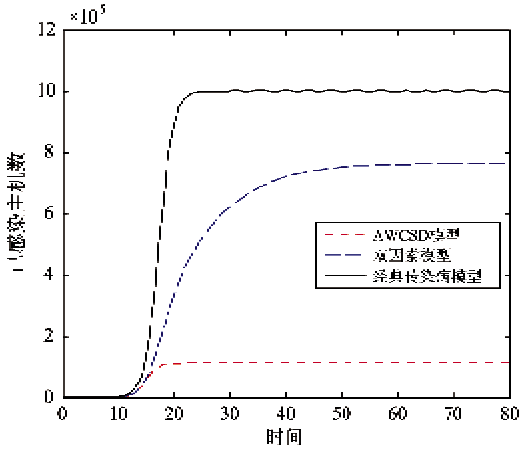


图 5　不同模型感染过的节点变化对比

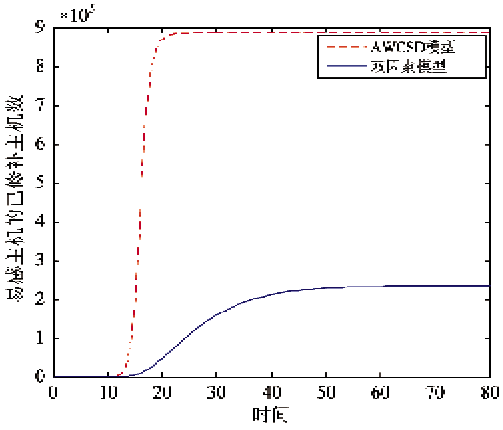


图 6　不同模型下修复节点的变化

3.2　感染主机初值的影响

网络中全部主机的数目 $N = 1000000$, $\eta = 3$, $\gamma_0 = 0.05$, $\mu_0 = 0.06/N$, $\tau = 2$, $\beta_0 = 0.8/N$, 检测与遏制服务器台数 $m = 100$ 时,在不同的感染主机初值($I_0 = 1, 10, 100, 1000$)的条件下,网络蠕虫正在感染的主机数和已经感染的主机数的变化见图7和图8。

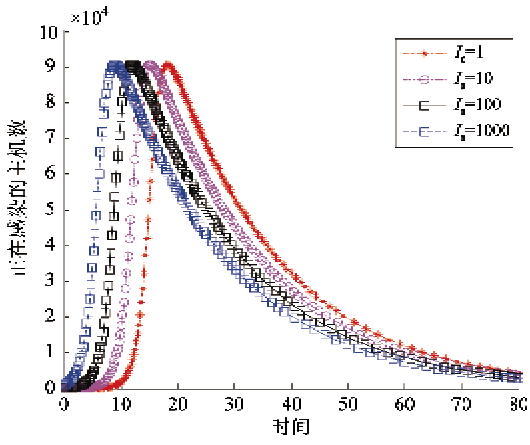


图 7　在不同初值 I_0 下 $I(t)$ 的变化

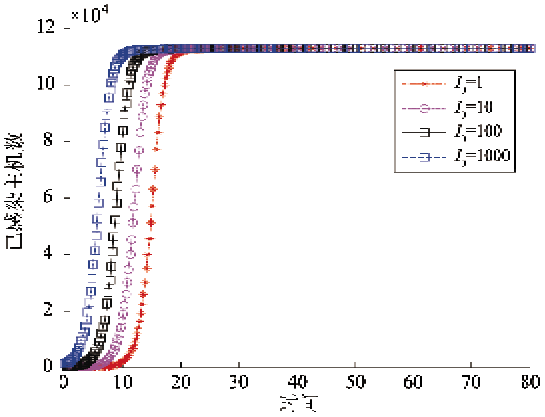


图 8 不同初值 I_0 下 $I(t)$ 的变化

从图中可以看出，在不同的感染主机数目的初值下，感染节点的初始值越大其达到峰值的时间越短，但是由于对抗算法是基于检测驱动的，因此会对感染的网络蠕虫进行远程清除免疫，同时触发自适应的远程修补会加快漏洞的修补速度，因此网络蠕虫的感染主机初值对模型的影响不大，都能够进行有效的遏制。

3.3 检测与遏制服务器台数的影响

网络中全部主机的数目 $N = 1000000$，$I_0 = 1$，$\eta = 3$，$\gamma_0 = 0.05$，$\mu_0 = 0.06/N$，$\tau = 2$，$\beta_0 = 0.8/N$，在不同的检测与遏制服务器台数（$m = 20,50,100,200,300,500$）下网络蠕虫的感染的主机数变化曲线见图 9。从图中可以看出，随着服务器台数的增加，网络蠕虫感染的数量越来越小，因此可以通过增加服务器的方法来增强遏制的效果。

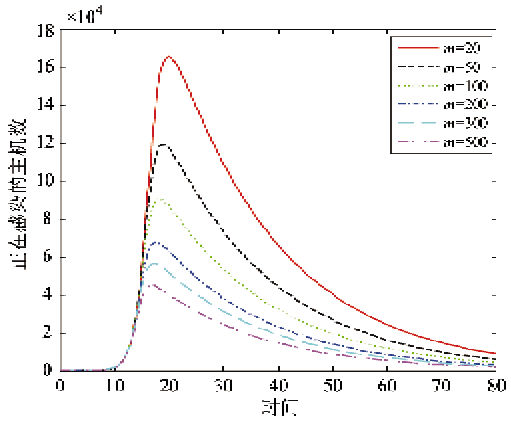


图 9 检测与遏制服务器台数 $I(t)$ 变化

3.4 远程清除与修补流量对网络的影响

由于进行主动修补补丁和远程蠕虫清除都会给网络带来流量负荷。在单位时间内由单台检测与遏制服务器产生的流量为 $s_1\sigma I(t)^{\tau}S(t) + s_2\phi I(t)$，$I(t) \in [0,N]$，$S(t) \in [0,N]$，其中 s_1 为主动修补补丁的大小，s_2 为远程网络蠕虫清除工具的大小，为计算方便，不妨设 $s_1 = s_2 = s$，s 为对抗工具。网络中全部主机的数目 $N = 1000000$，$I_0 = 1$，$\eta = 3$，$\gamma_0 = 0.05$，$\mu_0 = 0.06/N$，$\tau = 2$，$\beta_0 = 0.8/N$。检测与遏制服务器台数取 $m = 50 \sim 500$，$s = 4 \sim 10$k，绘制流量曲线见图 10。从图中可以看出，在检测与遏制服务器台数为 50，s 大小为 10k 时，流量最大值为44.961Mb/单位时间，这是因为蠕虫的远程清除是根据检测系统捕包侦听而获取感染地址的，不会对网络

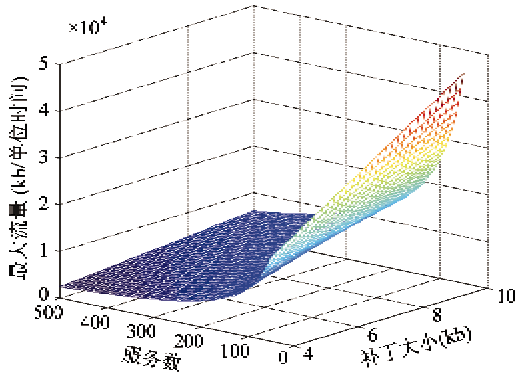
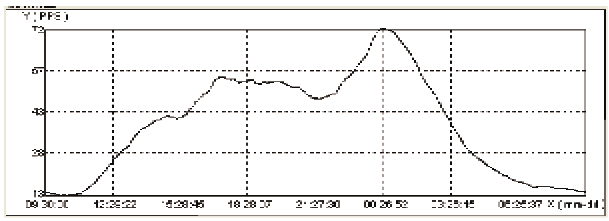
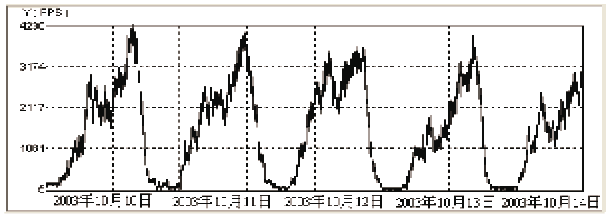


图 10 检测与遏制服务器产生的网络流量

造成流量冲击，而漏洞的远程修补是基于已知拓扑不会带来额外的流量，可见其对网络的影响是微小的。而良性网络蠕虫对恶性网络蠕虫进行遏制时，则会对网络造成很大的冲击，图 11（a）为正常情况下哈尔滨工业大学网络出口 ICMP 协议流量变化，图11（b）为Welchia蠕虫爆发时，出口ICMP



（a）正常状态



（b）Welchia 感染状态

图 11 正常情况下与 Welchia 感染时 ICMP 数据包流量的对比

数据流量变化图,其流量增长了70倍,可见Welchia作为遏制冲击波蠕虫的"良性蠕虫"对网络造成了很大的冲击。

4　结　论

本文在分析传统的蠕虫对抗技术及弱点的基础上,提出了利用漏洞来对抗蠕虫的新思路,这种方法能够克服传统被动的蠕虫对抗技术的弱点,能够修补网络上不可控节点的漏洞,同时清除网络感染节点的蠕虫。建立了基于检测驱动的网络蠕虫遏制模型并进行了理论分析,分析了感染节点的初值、检测与遏制服务器的台数、远程清除与修补流量等对网络的影响。通过与经典疫情模型和双因素模型的模拟对比证明了本文所建模型的遏制效果。实验结果表明,该模型能够在不增加网络流量的前提下对网络蠕虫的传播进行有效的遏制。

参考文献

[1] David M, Colleen S, Geoffrey M, et al. Internet quarantine: requirements for containing self-propagating code. In: Proceedings of the IEEE Conference on Computer Communications, San Francisco, USA,2003. 1901-1910

[2] Matthew M. Throttling virus: restricting propagation to defeat malicious mobile code. In: Proceedings of the 18th Annual Computer Security Application Conference, Las Vegas, USA, 2002. 61-68

[3] 方滨兴. 网络与信息安全研讨会专题发言. http://tech. sina. com. cn/focus/2005 net _ infosafe/index. shtml, 2005

[4] 汪立东. 操作系统安全评估与审计增强.[博士学位论文]. 哈尔滨:哈尔滨工业大学计算机学院, 2002. 42-47

[5] Eugene H S. The Internet Worm Program: An analysis: [Technical Report: CSD-TR-823]. http://spaf. cerias. purdue. edu/tech-reps/823. pdf ,1989

[6] 姜启源,谢金星,叶俊. 数学模型. 北京:高等教育出版社,2004. 135-144

[7] Frauenthal J C. Mathematical Modeling in Epidemiology. New York: Springer-Verlag, 1980. 3-15

[8] Wang Y, Wang C X. Modeling the effects of timing parameters on virus propagation. In: Proceedings of the ACM CCS Workshop on Rapid Malcode, Washington DC, USA, 2003. 61-66

[9] Zou C C, Gong Z, Gong W, et al. Code red worm propagation modeling and analysis. In: Proceedings of the 9th ACM Conference on Computer and Communications Security, Washington DC, USA, 2002. 138-147

[10] Chen Z, Gao L, Kwiat K. Modeling the spread of active worms, In: Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications, San Francisco,USA, 2003. 1890-1900

[11] Wang F W, Zhang Y K, Wang C Q, et al. Stability analysis of a SEIQV epidemic model for rapid spreading worms. *Computers & Security*,2010,29(4): 410-418

[12] Sumeet S, Cristian E, George V, et al. Automated worm fingerprinting. In: Proceedings of the 6th Conference of Symposium on Operating Systems Design & Implementation, San Francisco, USA, 2004. 4-4

Active worm countermeasure based on detection

Xin Yi[*,**], Zhang Qingpu[*], Yang Qinghai[**], Jin Dailiang[**]

([*]School of Management, Harbin Institute of Technology, Harbin 150001)

([**]Network and Information Center, Harbin Institute of Technology, Harbin 150001)

Abstract

According to the views that traditional countermeasure technologies are not sufficient to deal with the worm threat, and to defeat worms in an effective and timely manner, an effective worm countermeasure system must be established, this study presented an active worm countermeasure approach based on worm detection, which detects infectious hosts and scans susceptible hosts through the active worm countermeasure system based on detection (AWCSD), a worm detection system in the gateway of a network. The countermeasure system will clean the worms and patch the vulnerabilities in networks. The mathematic model of AWCSD was proposed based on the two-factor model, and then, the worm epidemic situation curves were depicted with different parameters.

Key words: worm, vulnerability, based on detection, active countermeasure