网络蠕虫的扫描策略分析*)

王方伟[1,2] 张运凯[2] 王长广[1,3] 马建峰[1]

(西安电子科技大学 CNIS 教育部重点实验室 西安 710071)[1]

(河北师范大学网络信息中心 石家庄 050016)[2] (河北师范大学物理科学与信息工程学院 石家庄 050016)[3]

摘 要 网络蠕虫已经严重威胁了网络的安全。为了有效防治网络蠕虫,首要任务必须清楚有什么扫描方法,以及这些扫描方法对蠕虫传播的影响。为此,本文构建了一个基于离散时间的简单蠕虫传播模型,通过对 Code Red 蠕虫传播的真实数据比较,验证了此模型的有效性。以此模型为基础,详细分析了蠕虫的不同扫描策略,如均匀扫描、目标列表扫描、路由扫描、分治扫描、本地子网、顺序扫描、置换扫描,并给出了相应的模型。

关键词 网络蠕虫,扫描策略,传播模型,恶意代码

Analysis and Research of Internet Worm Scanning Strategies

WANG Fang-Wei[1,2] ZHANG Yun-Kai[2] WANG Chang-Guang[1,3] MA Jian-Feng[1]

(The CNIS Key Laboratory of the Education Ministry, Xidian University, Xi'an 710071)[1]

(Network Center, Hebei Normal University, Shijiazhuang 050016)[2]

(College of Physics Science, Hebei Normal University, Shijiazhuang 050016)[3]

Abstract Internet worms have severely threatened Internet security. In order to effectively defend against them, it is very important to understand the kinds of scanning strategies and how the scanning strategies affect worm propagation. A simple worm propagation model based on a discrete time differential equation is provided. The different scanning strategies used by Internet worms, such as uniform scan, Hit-list scan, routable scan, divide-and-conquer scan, local subnet scan, sequential scan, permutation scan are analyzed in detail based on the model provided, and given their corresponding models.

Keywords Internet worm, Scanning strategies, Propagation model, Malicious code

1 引言

从网络蠕虫的"祖先"——Morris 蠕虫出现到现在已有 18 年时间,在此期间出现了大量的网络蠕虫及其变种,造成了重大的损失。影响比较大的蠕虫有:Code Red, Nimda, Slapper, Slammer, Blaster, Witty, Sasser,这从侧面说明现在的网络非常脆弱。Weaver[1] 提出了几种蠕虫快速传播策略,这更加重了防治的困难。为了能更好地防治网络蠕虫,就必须理解其属性(扫描策略、激活方式、传播途径等)和构建准确的传播模型,以便做到未雨绸缪,尽量变被动挨打为主动出击,减少蠕虫的影响面,降低其造成的损失。一般网络蠕虫由扫描、攻击、现场处理、复制四个模块组成。影响网络蠕虫传播速度的因素主要有三个:一是存在漏洞的主机被发现的速度;二是有多少潜在"脆弱"主机可以被利用;三是网络蠕虫对这些目标的感染速度。决定蠕虫传播速度的主要因素是对存在漏洞的主机被发现的速度,也就是能够找到多少有效的主机系统,这要由蠕虫的扫描模块完成。所以,扫描是网络蠕虫传播的前提条件。为了更快且更有效地防治网络蠕虫,必须从扫描策略入手。

Weaver[2] 讨论了蠕虫的传播策略,把传播策略分成五种:随机、外部的目标列表(Metaserver)、预先生成的目标列表(Hit-list)、内部目标列表(Topological)、被动,强调蠕虫的检测、分析和响应需要自动化,并认为对现有蠕虫的分析、数学建模和仿真是研究蠕虫的主要手段。一个精确的蠕虫传播模型有助于人们更清楚地认识蠕虫,有利于确定其在传播过程中的弱点并制定有效的防御措施,而且有利于评价蠕虫防御系统的性能。Staniford 等[3] 采用基于连续时间的简单传染病模型(SI)来分析均匀网络上蠕虫的扫描策略,并给出了一些防治蠕虫的措施。Cliff C. Zou 等[4] 也采用相同方法来研究扫描策略,并分别给出了扫描策略所对应的模型,最后提出了一个防御蠕虫的系统框架。这两个基于连续时间模型的缺点是:认为一经被扫描就立即具备感染其它主机的能力,而忽视了蠕虫本身代码大小的影响,因为代码在网络上传播也需要一定的时间(虽然很小)。Chen Zesheng 等[5] 和 Wang Yang 等[6] 分别提出了一个基于离散时间的蠕虫传播模型(SIS),并用此模型来模拟蠕虫的一些扫描策略,但共同特点是模型过于复杂;由于考虑了人为措施的影响,不能真实反映蠕虫的扫描策略在蠕虫传播过程中所起的作用。针对已有模型存在的问题,本文提出了一个兼有这两类模型优点的模型:(1)基于离散时间,计算机在没有被完全感染前不能感染其它主机,蠕虫可以同时感染同一目的主机;(2)基于 SI 模型,模型既简单又能较真实反映蠕虫扫描策略对蠕虫传播过程的影

*)国家 863 计划项目(2002AA143021)、河北省自然科学基金项目(F2006000177)、河北省教育厅科技基金(2004445)。**王方伟** 博士生,讲师,主要研究领域为网络信息安全;**张运凯** 博士,教授,主要研究领域为网络安全;**王长广** 博士生,副教授,主要研究领域为网络安全;**马建峰** 博士,教授,博士生导师,主要研究领域为密码学、信息安全、无线网络安全等。

响。

2 网络蠕虫的传播模型

互联网上真实蠕虫的传播是一个很复杂的过程。为了方便起见而又不失一般性，本文只考虑单一蠕虫，而且没有考虑网络拓扑限制。网络拓扑限制指感染主机不能直接感染任意易感主机，需要先感染一些中间主机才能到达目标主机，如邮件病毒(Melissa 和爱虫)是依赖于网络拓扑结构的，这部分本文不做讨论。

2.1 模型假设

为了既简明又能真实反映扫描策略的影响，对模型做如下假设：

(1)易感主机数 N 是一个常数，不随时间 t 的变化而变化，即没有新的易感主机进入系统；

(2)不考虑人为措施(如打补丁、隔离、断开网络等)和网络拥塞的影响，即蠕虫的扫描率 η 为常数；

(3)主机只有两个状态：易感 S 和感染 I，某一时刻只能处于其中一个状态，不能再次感染已经处于感染状态的主机，初始感染主机数为 $I(0)=I_0$；

(4)蠕虫需要一个时间单元完成感染过程。

2.2 网络蠕虫传播模型

虽然现在互联网地址没有完全连接，但均匀扫描蠕虫总能扫描整个地址空间 Ω，因为 IPv4 是 32 位地址，所以 $\Omega=2^{32}$。对于均匀的随机扫描方法来说，任意主机被扫描一次的概率为 $1/\Omega$。用 $S(t)$表示时刻 t 时的易感主机数(包括已经被感染的主机数)，用 $I(t)$表示时刻 t 时的已经被感染的主机数。在蠕虫开始传播之前($t=0$)时，$S(0)=N$，$I(0)=I_0$。

假设蠕虫的平均扫描率为 η，时刻 t 时的感染主机发出的扫描数为 $\eta I(t)$，那么在整个地址空间 Ω 内，任意 IP 地址被扫描一次的概率为 $1/\Omega$，所以在单位时间内平均每个感染主机扫描整个地址空间 Ω 内的一个特定 IP 地址的概率为 $p=(1-(1-1/\Omega)^{\eta I(t)})$。因为共有 $S(t)-I(t)$个易感主机，所以在下一时刻 $t+1$ 时，新增加的感染主机数为 $p(S(t)-I(t))$，即 $(S(t)-I(t))(1-(1-1/\Omega)^{\eta I(t)})$。在时刻 $t+1$ 时，已被感染的主机总数为 $I(t+1)=I(t)+(S(t)-I(t))(1-(1-1/\Omega)^{\eta I(t)})$。同时，$S(t+1)=S(t)=N$，所以蠕虫的传播模型为下式：

$$I(t+1)=I(t)+(N-I(t))\left(1-\left(1-\frac{1}{\Omega}\right)^{\eta I(t)}\right) \qquad (1)$$

其中 $t\geqslant 0$，且 $I(0)=I_0$

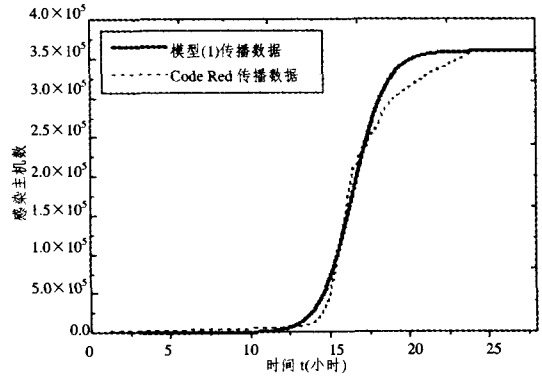

图 1 模型(1)的仿真结果同 Code Red 实际传播数据比较

本文采用 www.caida.org 网站上 Code Red 蠕虫实际感染主机的数目来验证此模型。在文[3]中连续时间模型的参数 $k=1.8$ 时，较好地拟合了 Code Red 的传播曲线，而 $k=\eta N/\Omega$，所以蠕虫的扫描率为 $\eta=k\Omega/N\approx 358$/min。在本模型中也采用这一数值，扫描地址空间为 Ω，$I(0)=1$，利用 Matlab Simulink7.0，所得结果如图 1 所示。

由图 1 可以看出，本模型曲线与 Code Red 实际传播的曲线还有一定的偏差，偏差为 3.5%。产生偏差的原因可能主要是采用固定的扫描率 η，而实际 Code Red 蠕虫扫描率是变化的，最大扫描率为 8500/min。但总的来说此模型是有效的，能够代表蠕虫的实际传播情况，下面就用模型(1)来仿真网络蠕虫扫描方法的性能。

3 网络蠕虫的扫描策略

扫描主机漏洞是蠕虫传播的前提，蠕虫传播经常用 ICMP Ping 包、TCP SYN、FIN、RST 和 ACK 包来进行探测[7]。影响蠕虫传播的主要因素是如何能快速找到新的目标主机，所以扫描方法的性能直接决定着蠕虫传播的速度。下面介绍几种网络蠕虫的扫描方法。

3.1 随机均匀扫描(Random Uniform Scanning)

随机扫描是感染蠕虫的主机在寻找新的攻击目标时不知道哪些主机系统有漏洞，随机地选择网上计算机进行扫描，如 Code Red 和 Slammer，所以扫描的目标为 IPv4 所有地址空间，即 $\Omega=2^{32}$。因此这类蠕虫可以用模型(1)来模拟。

3.2 基于目标列表的扫描(Hit-list Scanning)

Staniford 等[8]介绍了目标列表扫描方法。这种蠕虫在传播之前先搜集一些有漏洞的主机地址列表，传播时先感染这些地址列表中的主机，然后这些感染的主机再随机扫描互联网上的其它主机。扫描过程可分两个阶段：第一阶段是蠕虫感染 Hit-list 中的主机的过程，该阶段由于蠕虫编写者已经事先取得了存在漏洞的主机，而且这些机器都具有良好的网络连接，所以该过程速度极快，在开始几秒内就能完成。第二个过程因为没有先验知识，所以是随机传播，扫描地址空间为 Ω。所以，这种扫描方法也能用模型(1)来模拟，只不过初始感染主机数 $I(0)$为 Hit-list 表中的数量。

3.3 路由扫描(Routable Scanning)

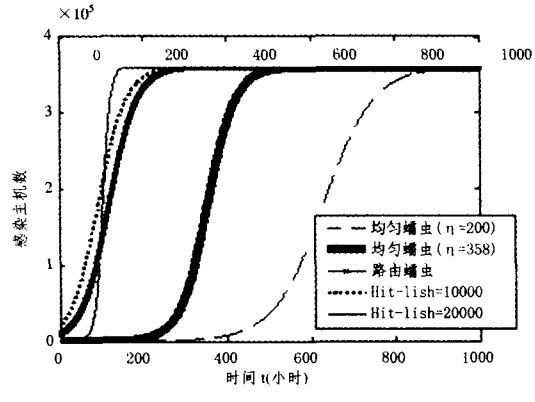

图 2 随机均匀、路由、目标列表扫描蠕虫的性能比较

Cliff C. Zou 等[9]介绍了路由扫描方法，这种网络蠕虫可以利用 BGP 路由前缀来减小蠕虫的扫描空间，也就是对 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。由于在 BGP 路由表中，只包含了 28.6%的 IPv4 地址空间，若采用路由扫描法，其探测地址空间将比随机扫描大约减小 3 倍，因此传播速度会增加。它的不足是蠕虫必须携带一个路由 IP 地址库，蠕虫代码比较大。这种方法也能用模型(1)来模拟，只不过扫描空间变为 0.286Ω。

为了比较这三种扫描方法的性能，采用相同的实验环境，扫描空间为 Ω、初始感染主机数 $I(0)=10$、易感主机数 $N=360000$、扫描率 $\eta=358$/min。结果如图 2 所示。

从图 2 可看出，扫描率 $\eta=200$ 的曲线最缓，在 780min 感染了大约 96%的易感主机；路由扫描蠕虫最陡，达到饱和状态时间为 110min，速度大约是前者的 7 倍。目标列表扫描蠕虫由于开始感染的主机数比较多，因此开始时曲线上升的最快。两个目标列表扫描蠕虫初始时感染的主机数相差 1 倍，但效果并不很明显(目标列表比较大的情况下)。在都感染 95%的情况下，时间相差 16.7min。扫描率对传播影响比较大，在其它参数相同时，达到饱和时间缩短了 410min。仿真结果说明：扫描率越大，目标列表越大，扫描空间越小，蠕虫传播越快；同时也给防治蠕虫提供了良好的对策：(1)采用虚拟地址，尽量不公开计算机地址，使得蠕虫初始可利用的目标列表减小；(2)虽然不能阻止蠕虫编写者利用这些公开的 BGP 路由表，但可以通过一定的措施限制编写者进一步降低蠕虫的扫描空间。

3.4 分治扫描(Divide-and-Conquer Scanning)

分而治之扫描的主要思想是：在释放蠕虫之前，作者需要收集一个 10000 到 50000 个有漏洞且网络连接良好主机的列表，在传播时，蠕虫给每个感染主机发送一个子列表，受害主机就按照子列表进行扫描。例如，主机 A 感染主机 B 后，主机 A 就把列表分成两部分，一半给主机 B，自己保留另一部分。即使在列表中只有 10～20%的主机有漏洞，这个快速分解方法也能很快通过列表且一分钟之内蠕虫能在全部有漏洞的主机上复制自身[3,4]。构造初始列表的常用技术有：秘密扫描、分布式扫描、DNS 搜索、Spiders 等，此扫描的特点是：隐蔽性强，通过逐渐分解列表，蠕虫越来越小；扫描造成的流量也进一步减小，更不容易检测。但是，如果分得列表的主机被关掉或死机，那这部分列表就会丢失，这种情况发生的越早，对蠕虫传播的影响越大。

假设每个感染主机均匀扫描自己地址空间的 IP 地址，由于没有先验知识，因此同一 IP 地址有可能被扫描多次。假设易感主机均匀分布在整个地址空间内，这样初始感染主机 $I(0)$中每个负责的扫描的地址数目就为 $\Omega/I(0)$，根据上面的分析，每个 $\Omega/I(0)$中只有一个感染主机。在时刻 t 时，共有 $I(t)$个感染主机，这样就把整个地址空间分成了 $I(t)$部分，每一部分负责扫描的地址空间为 $\Omega/I(t)-1$。因为易感主机是均匀分布的，蠕虫均匀选择其负责部分的某一 IP 地址，所以剩余的易感主机也呈均匀分布。这样每一感染主机平均要扫描的易感主机数为 $N/I(t)-1$。特定 IP 地址的被扫描的概率为 $p'=(1-(1-1/(\Omega/I(t)-1))^{\eta})$，所以在时刻 $t+1$ 时，新增加的感染主机总数为 $I(t)(N/(I(t)-1)p'$，同时，$I(t)\leqslant N\ll\Omega$，$\Omega-1(t)\approx\Omega$，所以分治扫描法的模型

$$I(t+1)\approx I(t)+(N-I(t))\left(1-\left(1-\frac{I(t)}{\Omega-I(t)}\right)\right)^{\eta} \qquad (2)$$

Flash 蠕虫采用目标列表和分治扫描来实现传播，是传播最快的蠕虫[3]。所采用试验参数为：Hit-list=10000，I(0)=10，扫描率 $\eta=358$/min。主机总数 N=360000，仿真结果如图 3 所示。

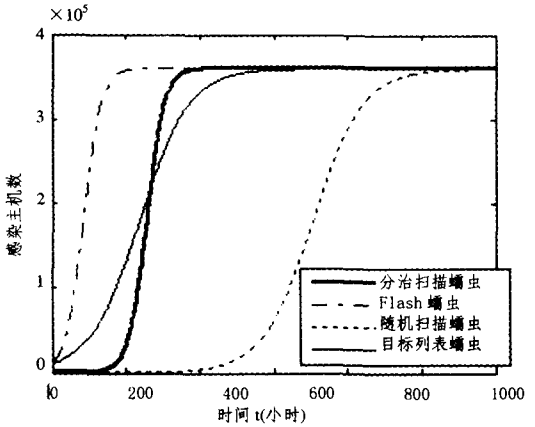

图 3 随机、目标列表、分治、Flash 蠕虫的性能比较

从图 3 直观地可以看出分治扫描蠕虫比随机和目标列表蠕虫传播的都要快，开始时传播比较慢是因为初始的感染主机数比目标列表蠕虫小得多，但增长幅度很大，在 135min 后就超过了目标列表蠕虫。Flash 蠕虫因兼有目标列表和分治蠕虫的优点，所以传播的最快，在 115min 时感染全部主机，而一般的随机扫描蠕虫要用 570min。

3.5 本地子网扫描(Local Subnet Scanning)

前面四种方法都假设易感主机在整个 IP 地址空间内均匀分布，然而实际情况并非这样，有的网络由于安装了一些保护措施(如防火墙、地址过滤、内容过滤等)使得蠕虫无法完成扫描，易感主机密度就比较小。有的网络缺乏保护，蠕虫可利用漏洞比较多，易感主机的密度就比较大，所以蠕虫编写者就实现了本地子网扫描方法：以比较大的概率扫描某一网段，再用比较小的概率完成随机扫描，这样做有利蠕虫快速感染一些的主机，又能跳出本网，从而感染更多的主机，如 Code Red II 蠕虫。

假设蠕虫以比较大的概率 q 扫描'/n'网段，以概率 $1-q$ 随机扫描。这样在 IPv4 网络中，每个'/n'网段有 2^{32-n} 个 IP 地址，假设蠕虫的扫描空间包含 K 个'/n'网段，即共有 $K2^{32-n}$ 个地址。每个网段有 N_k 个易感主机($k=1,\cdots,K$)，假设 $I_k(t)$为第 k 个网段在时刻 t 时的感染主机数，蠕虫扫描第 k 个网段内某一特定 IP 地址的概率就为 $p'=1-(1-1/2^{32-n})^{q\eta I_k(t)}$，扫描其它 $k-1$ 个网段内某一特定 IP 地址的概率为 $p''=1-[1-1/((K-1)2^{32-n})]^{(1-q)\eta I_k(t)}$，所以本地子网扫描的模型为

$$I_k(t+1)=I_k(t)+[N_k-I_k(t)]p'+\sum_{i\neq k}p''] \qquad (3)$$

其中初始感染主机数为 $I_k(0)$，$k=1,2,\cdots,K$。

实际网络中，易感主机并不是均匀分布在整个地址空间内，对于 A 类 IP 地址来说，可路由的网段不是 256 个，而是 116 个[9]。不失一般性，假设易感主机均匀分布在前 m($m<K$)个网段内，$N_1=N_2=\cdots N_m=N/m$，$N_{m+1}=\cdots=N_k=0$，而蠕虫编写者并不知道哪些网段没有易感主机，这样每个网段内蠕虫传播模型为

$$I_{k+1}(t+1)=I_k(t)=[p'+(m-1)p''][N_k-I_k(t)] \qquad (4)$$

又假设 $I_1(0)=I_2(0)=\cdots+I_m(0)=I(0)/m$，那么整体来说，蠕虫的传播模型为

$$I(t+1)=I(t)+\frac{[p'+(m-1)p'']}{m}[N-I(t)] \qquad (5)$$

为便于分析本地子网扫描蠕虫的性能，现同随机扫描蠕虫做比较。扫描空间为 Ω，初始感染主机数 $I(0)=10$，易感主机数 $N=360000$，扫描率 $\eta=358/\min$，结果如图 4 所示。

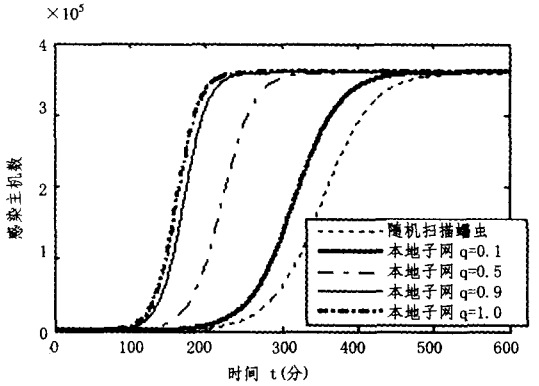

图 4　本地子网蠕虫和随机扫描蠕虫的性能比较

从图 4 可以明显地看出，本地子网扫描比随机扫描性能优越。本地扫描概率越 q 大，完成感染所需的时间越短，概率 $q=1$ 时传播最快。

3.6　顺序扫描(Sequential Scanning)

顺序扫描是指感染主机上的蠕虫按照一定顺序依次扫描自己所在网段内的地址。若蠕虫扫描的目标地址 IP 为 x，则扫描的下一个地址 IP 为 $x+1$ 或者 $x-1$。优点是对易感主机密度比较大网络有效，短时间内就可以感染大量主机；缺点是会产生大量的重复扫描，引起网络拥塞。假设感染主机 A 的 IP 地址为 x，采用 IP 地址递增方法，该蠕虫就从主机 A 开始依次扫描 $x+1, x+2, \cdots$，不失一般性，若 IP 地址为 $x+2$ 主机 B 被主机 A 感染，则主机 B 依次 $x+3, x+4, \cdots$。假设一个 C 类子网中所有主机都是易感主机，最坏情况下，扫描数为 $(1+255)*128=32768$，而实际可能只需要 255 次扫描就能感染所有主机，Blaster 是典型的顺序扫描蠕虫。

3.7　置换扫描(Permutation Scanning)

前面随机扫描方法严重缺陷是由于蠕虫没有先验知识不能确定哪些易感主机已经被感染，导致某些 IP 地址可能被扫描很多次，严重影响蠕虫的传播速度，如图 5(a)所示。如果蠕虫之间能协同工作，能及时交换已经感染主机的情况，那么理论上 IP 地址恰好被扫描一次，如图 5(b)所示，称这类蠕虫为理想蠕虫。

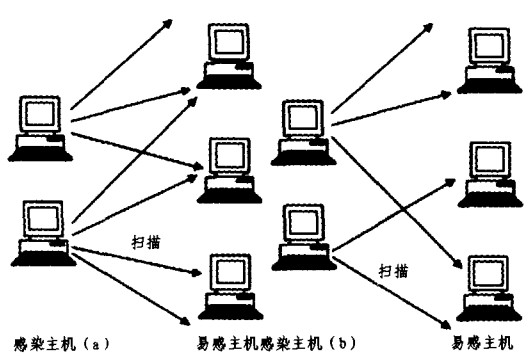

图 5　随机扫描蠕虫和理想扫描蠕虫的扫描方法比较

置换扫描方法[3]就是一个特例，它采用伪随机置换(Pseudo Ransom Permutation)来产生自调整的扫描，看上去有点像随机扫描。在置换扫描过程中，蠕虫共享 IP 地址空间的一个伪随机置换。用一个带有预先选好密钥的 32 位分组密码来产生置换，把一个 IP 地址加密就能得到在置换空间内的相应地址，反之，解密一个置换空间内的地址就能得到其 IP 地址，如图 4 所示。实际上置换扫描蠕虫是半理想蠕虫，它也执行随机扫描，只不过使无用扫描数量降到很低。这种方法比较复杂，不易实现。下面采用 D. H Lehmer 在 1948 年提出的线同余产生器(Linear Congruential Generator, LCG)来实现置换。其基本原理如下：

假设 X_i 是原始空间中的一个地址，X_{i+1} 为置换空间内相应的地址，X_{i+1} 和 X_i 间的关系为

$$X_{i+1}=X_i*a+b \bmod m \tag{6}$$

其中常数 $a=214013$，$b=2531011$，$m=2^{32}$，这样利用 LCG 就产生了区间$[0,2^{32}-1]$内所有整数的一个置换，如图 6 所示。

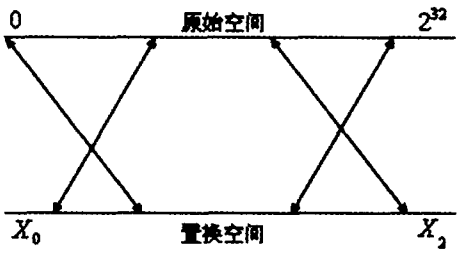

图 6　线性同余置换示意图

因为置换扫描浪费的扫描率极低，为了模拟其传播过程，假设感染主机间有完美的通信机制以保证每一个扫描对应一个地址，即每个地址只被扫描一次。虽然易感主机可能不是均匀分布，但经过置换后，可以近似地认为呈均匀分布。整个地址空间 Ω 被 N 个易感主机分成 N 块，每块大小为 Ω/N，每个感染主机在单位时间内可以新感染 $m=[1-(1-N/\Omega)^{\eta}]$ 个主机，所以在时刻 t 时，新增加的感染主机数为 $mI(t)$，所以置换扫描蠕虫的模型可以表示为

$$I(t+1)=I(t)+I(t)[1-(1-\frac{N}{\Omega})^{\eta}] \tag{7}$$

Warhol 蠕虫[3]采用目标列表和置换扫描相结合的方法，能在 15min 内感染互联网所有存在漏洞的主机，庆幸的是这类蠕虫到目前没有出现。为便于比较，本试验所采用的参数同文[3]一致：扫描率 $\eta=100/s$，易感主机总数 $N=300000$，所得模拟结果如图 7 所示。

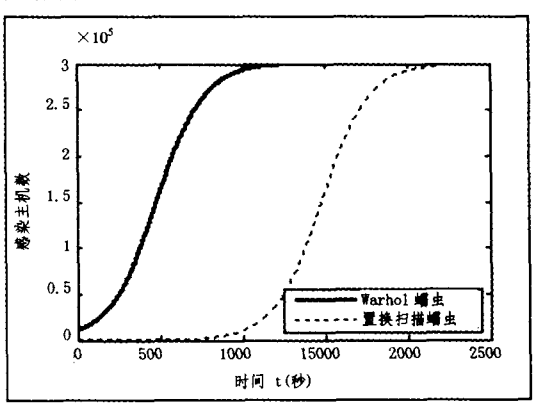

图 7　置换扫描蠕虫和 Warhol 蠕虫传播示意图

图 7 中置换扫描蠕虫传播曲线急剧上升，并迅速达到饱和状态，大约在 2000s 左右就感染了 99.5%的主机，说明采用

(下转第 158 页)

现;从过程实现角度,更复杂实用的例子需要去描述,另外借鉴AGM公理系统研究群体缩并也是需要引起重视的一个研究方向。

参考文献

1 ove Hansson S. Survey of non-prioritized belief revision. Erkenntnis,1999, 50: 413～427

2 Booth R. A negotiation-style framework for non-prioritised revision. In: Proceeding of the Eighth Conference on Theoretical Aspects of Rationality and Knowledge (TARK 2001),2001. 137～150

3 Borgida A, Imielinski T. Pecision making in committees: A framework for dealing with inconsistency and non-monotonicity In: Proceedings workshop on Nonmonotonic Reason,1984. 21～32

4 Alchourron C, Gardenfors P, Makinson D. On the logic of theory change: Partial meet contraction and revision functions. Journal of Symbolic Logic,1985,50:510～530

5 Konieczny S, Pino-perez R. On the logic merging. In: Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR'98), 1988. 488～498

6 Meyer T. On the semantics of combination operations. Journal of Applied Non-Classical Logics,2001, 11(1-2):54～89

7 Arieli O, Van Nuffelen B, Denecker M, Bruynooghe M. Coherent composition of distributed knowledge-bases through abduction. In: Proceeding of the Eighth International Conference on Logic for Programming, Artificial Intelligence and Reasoning(Lpar'01) LNCS, 2250, Springer, Berlin,2001. 624～638

8 Meyer T, Ghose A, Chopra S. Social choice, merging and elections. In: Proceedings of the Sixth Euripean Conference on Symbolic and Quantitative Approaches to Reasoning and Unvertainty (ECSQARU 2001),2001. 466～477

9 Arenas M, Bertossi L, Chomicki J. Consistent query answers in inconsistent databases. In: Proceedings of the Eighteenth ACM SIGACT SIGMOD SIGART Symposium on Principles of Database System (PODS'99),1999. 68～79

(上接第 108 页)

置换扫描的蠕虫效果显著。Warhol 蠕虫传播更迅速,在1158s(19.3min)感染了全部的易感主机,这给防治此类蠕虫带来了极大困难,人们不可能在这么短的时间内找到有效的措施。

结论及下一步工作 本文提出了一个基于离散时间的简单蠕虫传播模型,仿真试验表明此模型能较好地再现 Code Red 蠕虫的实际传播情况。利用此模型研究了几种扫描方法对蠕虫传播性能的影响,仿真表明置换扫描性能最好,分治扫描次之,随机扫描性能最差,并给出了一些防治措施。下一步工作包括:研究拓扑扫描、秘密扫描等方法,综合评价这些方法性能;研究这些扫描方法在 IPv6 网络上的性能;目前慢速蠕虫还没有有效的方法[10],将研究慢速蠕虫的传播模型及其防治方法。

参考文献

1 Weaver N. Warhol worms: the potential for very fast Internet plagues[EB/OL]. http://www.cs.berkeley.edu/~nweaver/warhol.html.

2 Weaver N. How many ways to own the Internet? Towards viable worm defenses [EB/OL]. http://www.cs.berkeley.edu/~nweaver/wormdefense.ppt. 2003.1

3 Staniford S, Paxson V, Weaver N. How to own the Internet in your spare time[A]. In: Proc. of the USENIX Security Symposium[C]. San Francisco, ACM Press, 2002.8.149～167. http://www.icir.org/vern/papers/cdc-usenix-sec02/cdc.pdf

4 Zou C C, Towsley D, Gong W. On the performance of Internet worm scanning strategies[J]. Performance Evaluation, 2006, 63(7): 700～723

5 Chen Z, Gao L, Kwiat K. Modeling the spread of active worms [A]. In: Proc. of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies[C]. Francisco, IEEE Press, 2003,4: 890～1900

6 Wang Yang, Chakrabarti D, Wang Chenxi, et al. Epidemic spreading in real networks: an eigenvalue viewpoint[A]. In: Proc. of the 22nd International Symposium on Reliable Distributed Systems (SRDS'03)[C], Florence, Italy, ACM Press, 2003.10: 25～34

7 Kephart J O, White S R. Measuring and modeling computer virus prevalence[A]. In: Proc. of the 1993 IEEE Symposium on Security and Privacy[C]. Oakland California IEEE Press, 1993.5: 2～15

8 Weaver N. Potential Strategies for High Speed Active Worms: A worst case analysis[EB/OL]. http://www.cs.berkeley.edu/~nweaver/worms.pdf, 2002

9 Zou C C, Towsley D, Gong W, et al. Routing worm: A fast, selective attack worm based on IP address information[A]. In: Proc. of 19th ACM/IEEE/SCS Workshop on Principles of Advanced and Distributed Simulation (PAD'05)[C], 2005. 199～206. http://www.cs.ucf.edu/~czou/research/routingWorm-PADS05.pdf

10 Nojiri D, Rowe J, Levitt K. Cooperative response strategies for large scale attack mitigation[A]. In: Proc. of the 3rd DARPA Information Survivability Conference and Exposition[C]. Washington DC, ACM Press, 2003,4:293～304