计算机病毒的随机传染模型

卢 勇,左志宏

(电子科技大学 计算机科学与工程学院,四川 成都 610054)

摘 要:传染是计算机病毒的主要特征之一,病毒的传染不仅提高了病毒的存活率,而且对计算机系统资源造成破坏和威胁。为了分析计算机病毒在计算机内的微观传染规律,在分析病毒传染机制的基础上,结合当前操作系统的特点,建立和分析了计算机病毒在单个计算机系统内的随机传染模型。得出结论:在单进程操作系统环境下,病毒的感染数量呈线性增长,感染强度相对稳定;在多进程操作系统环境下,病毒的感染数量和感染强度都呈 e 的指数级增长。最后提出了反病毒传染技术的发展趋势。

关键词:计算机病毒;随机传染;泊松过程;数学期望;模型;感染强度

中图分类号:TP309.5 **文献标识码**:A **文章编号**:1673-629X(2007)03-0172-04

Modeling Computer Virus Random Infection

LU Yong, ZUO Zhi-hong

(Sch. of Computer Sci. and Eng., Univ. of Electronic Sci. and Techn. of China, Chengdu 610054, China)

Abstract: Infection is one of the essential characteristics of computer viruses, which not only enhances subsistent probability of viruses, but damages and imperils computer resources. To study microcosmic infection of computer viruses, based on reseaching infectious mechanism of viruses and operating system characteristics, establishes and analyses the model of computer viruses random infection, which concludes that the number of virus infection increase linearly with steady infection intensity on single-process operating system, both the number of virus infection and infection intensity increase with e exponent on multi-process operating system. At last, brings forward some developmental trends against virus infection.

Key words: computer virus; random infection; Poisson process; expect; model; infection intensity

0 引 言

计算机病毒最早是由美国计算机病毒研究专家F. Cohen博士提出[1]。目前,关于计算机病毒比较权威的定义是:计算机病毒是一种在计算机系统运动过程中能把自身精确拷贝或有修改地拷贝到其他程序体内的程序[2]。计算机病毒的传染是指病毒在载体之间、系统之间的复制自身行为。可见,传染性是病毒在计算机系统中蔓延和持续存活的基础。下面以病毒的基本模型来进一步分析病毒的传染性。

病毒一般分成主控模块、传染模块、破坏模块和触发模块4个部分[2]:

```
病毒程序:=
{
    感染模块:=
    {循环:随机搜索一个文件;
        如果感染条件满足
            则将病毒体写入该文件;
        否则跳到循环处运行;}
    破坏模块:=
    {执行病毒的破坏代码}
    触发模块:=
    {如果触发条件满足
        返回真;
    否则返回假;}
    主控模块:=
    {执行传染模块;
     执行触发模块
        如果返回为真,执行破坏模块;
     执行原程序;}
```

病毒的传染模块主要完成病毒的自我复制,传染的一般过程[2]是:当病毒程序或染毒的程序运行时,病毒截取控制权,寻求感染突破口,当感染条件满足,即将病毒代码自制到宿主程序。病毒的感染条件根据不

收稿日期:2006-06-21

作者简介:卢 勇(1977-),男,湖北京山人,硕士研究生,研究方向为计算机病毒与信息安全;左志宏,副教授,研究方向为计算机网络与信息安全。

同的感染方式有不同的类型。例如常驻内存病毒一般修改系统中断,插入病毒中断程序,当某程序运行时,如果访问被病毒挂接的中断,则启动病毒中断程序,感染或破坏文件,否则运行原中断过程。许多文件性病毒在获得运行机会时,会随机搜索相关目录下的可感染文件,判断感染条件,例如是否存在感染标记、可重复感染的次数、文件类型等,条件满足则进行感染。不同类的病毒,甚至同类不同变种的病毒,它们的感染方式都有很大的不同。

1　计算机病毒的传染类别

根据病毒传染的途径不同,可将计算机病毒的传染分为单机传染、网络传染和人为传染。单机传染指病毒在单计算机系统内的传染过程;网络传染则是病毒通过网络在网络的各个结点间的传播过程;人为传染则主要是通过人的行为进行的病毒传播,如文件复制或磁盘拷贝等等。这里主要讨论单机传染。在单个计算机系统内,病毒的传染主要有以下两类:

(1)消极传染。这类病毒的传染处于非主动状态,传染过程主要取决于外界的行为。如常驻内存病毒,当它驻留内存时,如果外界程序的行为触发了它设置的条件,则病毒对其进行感染或破坏;否则该病毒不会进行传染。

(2)积极传染。这类病毒在获得运行机会时,会主动随机搜索相关文件,并进行感染。目前,大多数病毒的传染属于这种类型。

2　计算机病毒的随机传染模型

为研究病毒在单个计算机的传染行为,以典型的积极传染性病毒为例,建立了相应的随机传染模型。令 $X(t)$ 为某时刻 t 随机感染文件数,初始环境如下:

(1) 系统中有 N 个可被感染文件,这 N 个文件在运行时具有相同的优先级。初始状态时 N 个文件中仅有一个病毒文件,并且获得运行机会,此时系统中已经存在 W_0 个进程。

(2) 考虑到操作系统时间片轮转的调度策略(时间片为 S_t),以病毒完成对文件一次传染的时间 $\triangle(v)$ 为时间单位,将病毒的整个传染过程时间离散化。假设 $\triangle(v)$ 为 S_t 的倍数,即 $\triangle(v) = kS_t$,k 为正整数。如果系统中除染毒进程外有其他进程,则在 $\triangle(v)$ 后,系统进行进程切换。

(3) 感染方式为单次感染,病毒在对 N 个文件的传染过程中,触发条件不满足,不运行破坏模块。

(4) 考虑在感染过程的某一时刻用户可能产生新的进程,假设:$N-1$ 个文件中的任一文件在任何时刻都可能运行;在两个不相交的时间区间内,随机选择的运行文件数目相互独立;在 $\triangle(v)$ 内,某个文件运行的概率正比于 $\lambda\triangle(v)$,并且在 $\triangle(v)$ 内出现两个或两个以上文件运行的概率趋于 0,其中 λ 为常数,表示单位时间内随机选择文件的平均数目。令 $N(t)$ 为某一时刻 t 在 $N-1$ 个文件中的文件运行数目,则 $\{N(t), t>0\}$ 是泊松过程[3]。

$$P\{N(t)=x\}=\frac{(\lambda t)^x}{x!}e^{-\lambda t}\quad x=0,1,2,3\cdots$$

(5) 外界用户生成的新进程有两种类型:染毒进程和未染毒进程。染毒进程产生的概率等于 N 个文件中带毒文件数与总文件数 N 之比。

设 $P(s)$ 为 s 时刻产生染毒进程的概率,则:

$$P(s)=\frac{X(s)}{N}$$

其中 $X(s)$ 为 s 时刻系统中已经被感染的文件数,N 为总文件数。由于外界用户生成新进程的过程是泊松过程,并且生成的新进程分两种类型,令 $N_1(t)$ 为 t 时间内染毒进程生成的数目,$N_2(t)$ 为 t 时间内非染毒进程生成的数目,根据泊松过程的分流原理[3],其相应的数学期望为:

$$E(N_1(t))=\lambda pt$$

$$E(N_2(t))=\lambda(1-p)t$$

其中 p 为:

$$p=\frac{1}{t}\int_0^t P(s)\mathrm{d}s=\frac{1}{t}\int_0^t\frac{X(s)}{N}\mathrm{d}s$$

假设 ξ_i 为第 i 个非染毒进程的运行时间。由于程序的运行时间是一随机变量,这里考虑最坏情况下的感染情形:

(1) 初始状态时系统的 N 个文件中只有染毒程序运行,即 $W_0=0$,并且不考虑系统进程的影响。

(2) 在感染过程中若外界用户随机产生的进程为染毒进程,则其一直运行到 N 个文件被完全感染为止。

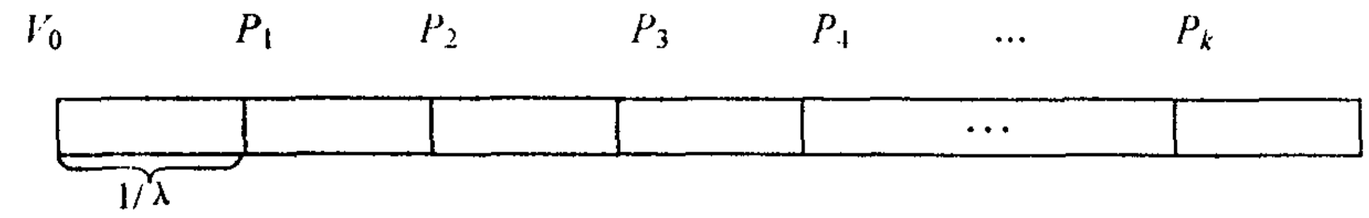
(3) 新染毒进程产生时忽略系统对它的初始化时间,及染毒进程运行被感染的病毒代码。

考虑到总的时间 t 为染毒进程的运行时间和非染毒进程的运行时间之和,因此

$$X(t)=\begin{cases}1 & t=0\\ \dfrac{1}{\Delta(v)}\left(t-\sum_{i=0}^{E(N_2(t))}\xi_i\right)+X(0) & t>0\end{cases}$$

ξ_i 是相互独立的随机变量,影响的因素很多,例如程序时间复杂度、用户提前停止程序、系统性能等等,令 $E(\xi)$ 为随机产生的非染毒进程的平均运行时间。由于泊松过程 $\{N(t), t>0\}$ 的时间间隔服从参数为 λ

的指数分布，其数学期望为 $1/\lambda$[3]，如下图所示。



图中 V_0 为初始状态运行的染毒进程，$P_1, P_2, P_3, P_4, \cdots, P_k$ 为依次随机产生的第一个进程、第二个进程 … 第 k 个进程，其平均时间间隔为 $1/\lambda$。令 $T = (N-1)\triangle(v)$ 为完成 $N-1$ 个文件的传染所需要的总的时间，若在 $0 \sim 1/\lambda$ 时间内 $T \leqslant 1/\lambda$，即对 N 个文件传染的总时间小于 $1/\lambda$，则 $E(\xi) = 0$；否则，当随机产生的进程都为非染毒进程，并且产生后一直运行到 N 个文件都被感染，此时 $E(\xi)$ 最大。即

$$E(\xi) = \begin{cases} 0 & \lambda \leqslant \dfrac{1}{T} \\ \dfrac{t-T}{\lambda t} & \lambda > \dfrac{1}{T} \end{cases}$$

将 $E(\xi) = \dfrac{t-T}{\lambda t}$ 变换为 $\dfrac{T}{\lambda t} = \dfrac{1}{\lambda} - E(\xi)$，由 $\dfrac{1}{\lambda} - E(\xi)$ 得到 $E(\xi)$ 的大致范围为：

$$E(\xi) = \begin{cases} 0 & \lambda \leqslant \dfrac{1}{T} \\ \left(0, \dfrac{1}{\lambda}\right) & \lambda > \dfrac{1}{T} \end{cases}$$

因此：

$$X(t) = \begin{cases} \dfrac{1}{\Delta(v)}[t - \lambda t(1-p)E(\xi)] + X(0) & t > 0 \\ 1 & t = 0 \end{cases}$$

代入 p：

$$X(t) = \begin{cases} \dfrac{t - t\lambda E(\xi)}{\Delta(v)} + \dfrac{\lambda E(\xi)}{\Delta(v)N}\int_0^t X(s)\mathrm{d}s + X(0) & t > 0, \lambda > \dfrac{1}{T} \\ \dfrac{t}{\Delta(v)} + 1 & t \geqslant 0, \lambda \leqslant \dfrac{1}{T} \end{cases}$$

求解方程得：

$$X(t) = \begin{cases} \dfrac{t}{\Delta(v)} + 1 & t \geqslant 0, \lambda \leqslant \dfrac{1}{T} \\ \dfrac{\lambda E(\xi) + N - N\lambda E(\xi)}{\lambda E(\xi)} e^{\frac{\lambda E(\xi)}{\Delta(v)N}t} + \dfrac{N\lambda E(\xi) - N}{\lambda E(\xi)} & t > 0, \lambda > \dfrac{1}{T} \end{cases}$$

以上为积极传染性病毒在单个计算机内的传染模型，其感染强度（单位时间内感染的文件数目）为：

$$X'(t) = \begin{cases} \dfrac{1}{\Delta(v)} & t > 0, \lambda \leqslant \dfrac{1}{T} \\ \dfrac{\lambda E(\xi) + N - N\lambda E(\xi)}{\Delta(v)N} e^{\frac{\lambda E(\xi)}{\Delta(v)N}t} & t > 0, \lambda > \dfrac{1}{T} \end{cases}$$

可以看出，积极传染性病毒在计算机内的传染总体上与文件总数 N、病毒完成单次传染的时间 $\triangle(v)$、事件发生率 λ、非染毒进程运行的平均时间 $E(\xi)$ 相关，其中 N 与 $\triangle(v)$ 是前提因素，λ 与 $E(\xi)$ 是用户的随机因素，这些因素都是不可控因素，基于多任务操作系统，如 WINDOWS, UNIX, LINUX，当 $\lambda \leqslant 1/T$ 时，积极传染性病毒随时间感染文件的数量以变化率 $1/\triangle(v)$ 直线增长；当 $\lambda > 1/T$ 时，积极传染性病毒感染文件的数量与感染强度都随时间以 e 的指数级进行增长。由于传染的高效性和不可控因素的影响，病毒的微观传染具有强不可控性。

为了验证上述模型，采用离散事件模拟方法，得到的模拟曲线如图 1、图 2 所示（实验环境：操作系统 Windows 2000，CPU 2.00GHz，512M RAM；$\triangle(v)$ 时间单位：μs，其他时间单位：s；$N = 10000, \lambda = 0.3, \triangle(v) = 500\text{ms}, E(\xi) = 0.05\text{s}$）。

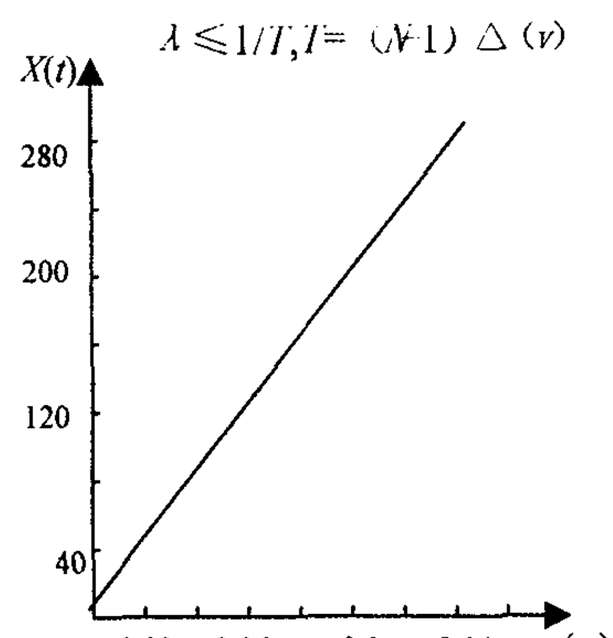


图 1 单进程环境下病毒的传染结果图

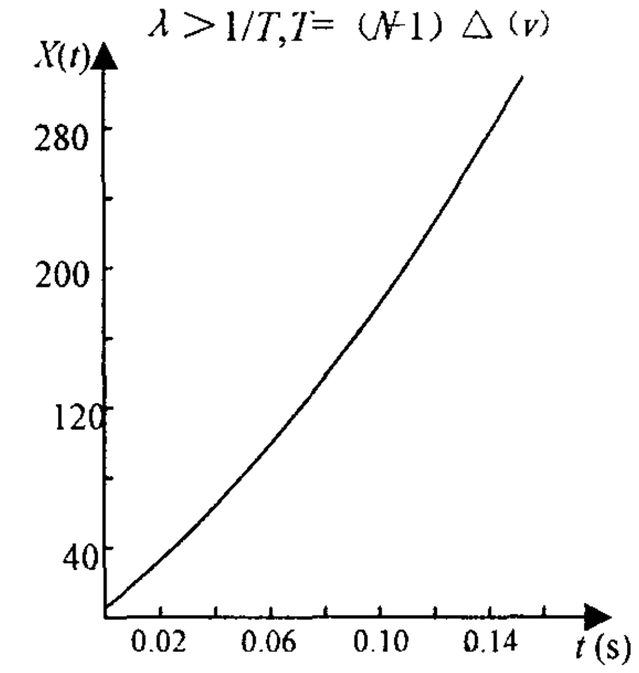


图 2 多进程环境下病毒的传染结果图

当 $\lambda \leqslant 1/T, T = (N-1)\triangle(v), t > 0$ 时，即完成对 N 个文件的传染时间在 $\{N(t), t > 0\}$ 泊松过程的时间间隔之内，此时 $X(t)$ 正比于 $t/\triangle(v)$，如图 1，其感染强度为 $1/\triangle(v)$，即完成单次传染的时间越短，单位时间内感染的文件数就越多。当 $\lambda > 1/T, T = (N-1)\triangle(v), t > 0$ 时，即完成对 N 个文件的传染时间大于 $\{N(t), t > 0\}$ 泊松过程的时间间隔，则在传染过程中有新的进程产生，传染曲线如图 2，其感染强度随时间以 e 的指数级增长，随时间的增加，感染速度越来越快。

3　反病毒传染分析

从上面的随机感染模型和实验可以看出,一旦计算机内有病毒并且获得运行机会,在很短的时间内会感染大量的用户资源,因此,如何控制病毒传染,或者病毒进行有限次传染后及时进行清除是反病毒领域面临的重要课题。对于已知病毒,病毒进行传染前能被反毒软件监测并清除。对于未知病毒,抑制病毒传染和及时检测出病毒相当困难,当前处于反病毒前沿广泛运用的技术,如虚拟机技术、启发式扫描、行为检测技术等等[4],也不能很好地解决对未知病毒的及时发现和清除问题。

消极传染性病毒传染的一般的方式是驻留内存、待机感染,这种病毒具备一定的隐蔽性,由于它对文件的感染主要取决外界程序的行为,相比积极性病毒传染,感染一定数目的文件时间较长。及时对内存扫描、检测和清除是对付这种病毒的主要手段[5]。这种病毒在目前的平台已经很少,取而代之的是驻留内存后采用积极传染的方式进行感染。相比消极传染,积极性传染的效率极高。由前面的结论可知,在当前计算机系统下,未知病毒的微观传染在用户态具备不可控性,反病毒传染需要向更微观的方向发展。未来的反病毒传染技术可以考虑建立在操作系统级,由操作系统监控程序运行,一旦发现程序对其他文件的自我复制行为,进行特征标记,当程序自动对其他文件的写操作超过一定数目时,由判断此程序可能为病毒,并做出相应措施,或交给用户处理,或由操作系统自身处理。

近年出现的内建 CPU 反病毒技术如 INTEL 的 EDB 和 AMD 的 EVP,能够对各种程序进行监视,阻止病毒在受保护的内存位置运行有害代码,是反病毒技术向更微观层次转变的趋势。

4　结束语

抑制传染是反病毒的主要手段之一,随着网络给计算机病毒传播带来的巨大便利,今后计算机系统面临更多病毒入侵的挑战。在保持现行计算机体系结构的前提下,深入分析计算机病毒的规律和发展趋势,克服操作系统的脆弱性,研究软硬件结合的反病毒技术,对反病毒领域的发展将起到重大的推动作用。

参考文献:

[1] Cohen F. Computational aspects of computer viruses[J]. Computers & Security, 1989,8(4):325－344.

[2] 傅建明,彭国军,张焕国. 计算机病毒分析与对抗[M]. 武汉:武汉大学出版社,2004.

[3] 张　波,张景肖. 应用随机过程[M]. 北京:清华大学出版社,2004.

[4] Szor P. The art of computer virus research and defense[M]. [s.l.]:Addison Wesley Professional,2005.

[5] Ford R. The future of virus detection[J]. Information Security Technical Report, 2004,9(2):19－26.

(上接第 171 页)

的方法,保证除了信任站点之外没有其他实例存取移动 Agent。因此,没有不信任的第三方能复制、改变、销毁移动 Agent。如果其中的某个信任站点攻击了 Agent,则登记的日志信息可以跟踪违反的主机,并确认责任。

4　结束语

移动 Agent 的可靠性与安全性问题是移动 Agent 研究的重要内容。文中从移动 Agent 迁移的角度,提出利用信任站点、基于分布式事务等机制的移动 Agent 迁移机制。该机制能很好解决由于网络的不可靠性等因素造成移动 Agent 在迁移过程中可能产生的失效等问题,提供了安全性与透明容错支持。

参考文献:

[1] 陶先平,吕　建,张冠群,等. 一种移动 Agent 结构化迁移机制的设计和实现[J]. 软件学报,2001,11(7):918－923.

[2] 张冠群, 陶先平,李　新,等. Mogent 系统迁移机制的设计与实现[J]. 计算机研究与发展,2001,38(9):1035－1041.

[3] 黄少寅,尹长青,高传善,等. 移动代码加密理论研究[J]. 计算机研究与发展,2003,40(11):1626－1634.

[4] Pleisch S, Schiper A. Approaches to Fault and Transactional Mobile Agent Execution－An Algorithmic View[J]. ACM Computing Surveys,2004,36(3):219－262.

[5] Roth V. Programming satan's Agents[C]// In Proc of the 1st Int Workshop on Secure Mobile Multi－Agent Systems. Montreal, Canada:[s.n.], 2001.

[6] Pleisch S, Schiper A. Modeling fault－tolerant mobile Agent execution as a sequence of agreement problems[C]// In Proc. of 19th IEEE Symposium on Reliable Distributed Systems (SRDS'00). Nuremberg, Germany:[s.n.],2000:11－20.

[7] 吕　建,陶先平,马晓星,等. Mogent 系统的设计、实现与应用研究[C]//全国软件与应用学术会议论文集. 南京:南京大学出版社,2005.

[8] 赵　靖,程　欣,崔　刚,等. 容错的移动代理框架及执行[J]. 计算机工程与应用,2005(1):144－147

[9] 李唯唯. 基于移动 Agent 的电子商务交易协商方案[C]// 全国软件与应用学术会议论文集. 南京:南京大学出版社. 2005.