文章编号：2096-1618(2017)01-0008-06
稿件来源：科学网
网　　址：http://blog.sciencenet.cn/blog-453322-1001684.html
发表时间：2016-09-08

安全通论(14)

——病毒式恶意代码的宏观行为分析

杨义先，　钮心忻
(北京邮电大学信息安全中心，北京 100876)

摘要：在网络空间安全领域，黑客四处“点火”；红客则疲于奔命，忙于“救火”。由于始终被动，所以，红客笃信：远水救不了近火。但是，事实却是：欲救某些“五味真火”，还真得依靠观音菩萨，从遥远的西天带来玉净瓶，用杨柳枝泼洒仙脂露。本文便从遥远的生物医学领域，带来传染病动力学之“远水”，试图来救病毒式恶意代码这盆“近火”。将传染性疾病防控的一些经典思想和理念，引入网络空间安全保障体系建设之中。

doi：10.16836/j.cnki.jcuit.2017.01.002

0　引言

人类一直与各类疾病(特别是瘟疫等传染病)作斗争；而且，至少在三百多年前，徐光启就将数学手段引入了生物统计。如今，动力学理论，这一数学分支，已被广泛应用于国内外生物医学领域，并取得了若干重大成果，比如，催生了多位诺贝尔奖获得者。

从外观和形态上来看，网络空间安全与人体疾病很相似，甚至安全界的不少名词(比如，病毒、免疫力、传染等)都是从医学中借用过来的。但是，也许是因为网络安全专家太忙，也许是因为数学门槛太高，也许是因为历史还不够久远等，总之，生物医学家和安全专家至今仍然是“各唱各的调，各吹各的号”。如果没有人牵线，也许他们永远都会“比邻若天涯”。本文不才，愿做无偿媒婆，将生物医学(特别是生物数学)中的若干经典成果和思路，介绍给网络安全专家。若能促成医生和安全专家的此桩姻缘，也不枉丘比特借箭一回；若能吸引一批生物数学家进入网络安全领域，那就更好了。

恶意代码是最头痛的安全问题之一，它甚至是整个软件安全的核心。虽然单独对付某台设备上的指定恶意代码并不难，但是，网上各种各样的海量恶意代码，却像癌细胞一样，危害着安全，而且，既杀之不绝，又严重消耗正常体能。过去，安全专家在对付恶意代码的微观手段方面，做了大量卓有成效的工作；可是，在宏观手段方面，几乎无所作为，这正是我们要向医生学习的地方。

本文以安全专家为读者对象，为保持完整性，先简要归纳恶意代码的基础知识。

狭义上说，恶意代码是指故意编制或设置的、会产生威胁或潜在威胁的计算机代码(软件)。最常见的恶意代码有计算机病毒、特洛伊木马、计算机蠕虫、后门、逻辑炸弹等。当然，广义上说，恶意代码还指那些没有作用却会带来危险的代码，比如，流氓软件和广告推送等。本文重点考虑狭义恶意代码。

恶意代码的微观破坏行为，表现在许多不同的方面，比如，口令破解、嗅探器、键盘输入记录，远程特洛伊和间谍软件等等。黑客利用恶意代码便可能获取口令，侦察网络通信，记录私人通信，暗地接收和传递远程主机的非授权命令，在防火墙上打开漏洞等。

恶意代码的入侵手段主要有三类：利用软件漏洞、利用用户的误操作、前两者的混合。有些恶意代码是自启动的蠕虫和嵌入脚本，本身就是软件，它们对人的活动没有要求。而像特洛伊木马、电子邮件蠕虫等恶意代码，则是利用受害者的心理，操纵他们执行不安全的代码。还有就是哄骗用户关闭保护措施来安装恶意代码等。

恶意代码的主要传播方式是病毒式传播，即，某台设备被恶意代码击中后，该受害设备又将再去危害其它设备。当然，也并非所有恶意代码都采取这种病毒式传播。为严谨起见，本文所说的恶意代码，都已经暗含病毒式传播的假设。

恶意代码也像病菌一样，千变万化不断升级，其演化趋势表现在：种类更模糊、混合传播模式越来越常见、多平台更加多样化、欺诈手段(包括销售技术等)

更普遍、更加智能化、同时攻击服务器和客户端、对操作系统(特别是 Windows)的杀伤力更大、类型变化越来越复杂等。不过,幸好本文只关注宏观行为,所以,恶意代码的微观变种可以忽略不计。

特别提醒:本文的思路、方法和结果,对以病毒式恶意代码类似的所有破坏行为都是有效的,只是为了不分散读者的注意力,我们才限用了“病毒式恶意代码”这个名词。比如,谣言的传播就具有典型的病毒特性,谣言的上当者,通常又会有意或无意地去传播谣言,为害别人。

1　恶意代码的动力学态势分析

以下所有分析,都基于这样一个已知的数学事实:一切随空间和时间变化的量的数学,都属于偏微分方程领域!

按照安全行业的习惯,下面我们不加区别地使用设备、用户、人、终端等名词;同时中毒、感染、传染、受害、伤害、被攻击、被击中等词也都互相通用。

1.1　死亡型恶意代码的态势

考虑这样一类恶意代码:它给你造成不可挽回的损失(比如,获取了你的银行卡密码并取走你的钱等)后,再以你的身份去诱骗你的亲朋好友;如此不断为害下去。由于它们造成的损失不可弥补,所以,称其为“死亡型”,相当于某人染 SARS 病毒死亡后,会继续传染身边人员一样。(有些“不转死全家”的谣言,也可看成这样的死亡型恶意代码)

设网络的用户数为 N,在 t 时刻,已经受害的用户数为 $T(t)$,暂未受害的用户数为 $S(t)$,那么,有恒等式 $S(t)+T(t)=N$。再令 $f(S,T)$ 为在“已有 T 人受害,S 人暂未受害”条件下,受害事件发生率,于是,有下面两个微分方程

$$\mathrm{d}T(t)/\mathrm{d}t=f(S,T) \text{ 和 } \mathrm{d}S(t)/\mathrm{d}t=-f(S,T)$$

在生物医学的流行病学中,有一个可借鉴的概念,传染力 $\lambda(T)$,它表示在已有 T 台设备中毒的情况下,暂未中毒的设备与中毒者相连接的概率,所以,$f(S,T)=\lambda(T)S$。还有另一个概念,即,传染率 β,它表示一个未中毒设备在连接到中毒者后,被传染的概率;所以,$\lambda(T)=\beta T$。于是,$f(S,T)=\lambda(T)S=\beta TS$,即,它是一个双线性函数。此种近似,已经过医学中的长期实际数据检验,准确度足够高;而对比病毒式恶意代码和人类疾病的传播,它们的传染特性并没有明显差别,所以,我们得到如下微分方程

$$\mathrm{d}T(t)/\mathrm{d}t=\beta T(N-T)$$

它的解析解为 $T(t)=NT(0)/\{T(0)-[T(0)-N]e^{-\beta Nt}\}$,这就是 t 时刻的受害者人数,其中 $T(0)$ 表示初始受害人数。注意到,只要 $T(0)>0$(即,刚开始时至少有一个受害者),那么,当 $t\to\infty$ 时,就一定有 $T(t)\to N$(全体用户数),所以,面对死亡型恶意代码,如果大家都旁观,不采取任何防护措施的话,最终将全体死亡,即全体受害。虽然现实中,大家不可能都只旁观,但是,这个理论结果也警告我们:网络安全,人人有责。

一旦大家都重视安全,并采取了各种事前预防和事后抢救的措施后,将出现下面 1.2 节中的康复型恶意代码。

注:此节和下面的许多分析中,我们在建立模型时都采用了诸如传染率、发生率等线性简单模型。有些读者可能会觉得这不够精确,对此我们解释如下:(1)由于在现实中,根据真实的原始统计数据,人类本来就只习惯于给出这些简单且形象的各种比率,所以,在建立微分方程模型时,也只好利用这些比率(比如,此处就取 $f(S,T)=\lambda(T)S=\beta TS$),否则,就变成了“为数学而数学”的游戏了;(2)线性(或双线性)微分方程组,相对来说,容易求出精确的解析解(其实有时也很难),并由此可以进行更深入的分析;(3)已有大量医学数据,对此类模型的准确性进行了长期的正确性验证。

1.2　康复型恶意代码的态势

与死亡型不同,在康复型模型中,受害用户在经过救治后,又可以康复成为暂未受害的用户,当然,该用户也可能再次受害。其实,绝大部分恶意代码,特别是诱骗类恶意代码,都是这种康复型的,比如,当你的电脑中毒崩溃后,你至少可以重新格式化嘛,当然,随后又可能再次崩溃,实际上每个人的电脑可能都不止崩溃过一次。(绝大部分人对谣言的反应,也等同于这种康复型恶意代码,因为,辟谣后,大部分人都会再次被谣言欺骗)

此时,除了 1.1 节中的 S、T、$F(S,T)$ 和 N 等概念外,再引入另一个概念,即 $g(T)$,它表示在 T 个受害者中,有 $g(T)$ 个用户被康复成正常健康用户,从而,变成暂未受害用户。若用 γ 表示康复率(生物医学经验告诉我们:每个受害者,在下一小段时间 δt 内,被康复的概率为 $\gamma\delta t+0(\delta t)^2$。并且受害者被康复的时间,服从均值为 $1/\gamma$ 的指数分布。)那么,$g(T)=\gamma T$。所以,有微分方程组:

$dT(t)/dt=f(S,T)-g(T)=\beta TS-\gamma T$ 和

$dS(t)/dt=-f(S,T)+g(T)=-\beta TS+\gamma T$

若令 $u(t)=S(t)/N$, $v(t)=T(t)/N$, $t'=\gamma t$ 和 $R_0=\beta N/\gamma$,那么,上面的两个微分方程就变为

$du/dt=-(R_0u-1)v$ 和 $dv/dt=(R_0u-1)v$,

其定义域为 $D=\{0\leqslant u\leqslant 1,0\leqslant v\leqslant 1,u+v=1\}$。

注意,这里的 R_0 是一个很重要的参数,其含义可以进一步解释为:$R_0=(\beta N)/\gamma$,它的分子部分表示一个中毒设备与 N 个健康设备之间的有效接触率(若无接触,当然就不可能被感染),分母部分 $1/\gamma$ 是受害者的平均染病周期,所以,R_0 是一个受害者在一个染病周期内,平均传染的设备个数。根据 R_0 的取值情况,可以得到如下重要结论,

定理 1:针对康复型恶意代码,如果 $R_0<1$,那么,康复型恶意代码就会最终被消灭,即,无人受害;反过来,如果 $R_0>1$,那么,康复型恶意代码就会在一定范围内长期为害,具体地说,受害者人数将长期徘徊在 $N(1-1/R_0)$ 附近。

证明:由于 $dv/dt=(R_0u-1)v<(R_0-1)v$,所以,如果 $R_0<1$,那么就有 $dv/dt<0$,即,受害者人数不断地严格减少,最终当然趋于零,从而,该恶意代码被消灭。

另一方面,如果 $R_0>1$,则由于 $u+v=1$,所以,微分方程 $dv/dt=(R_0u-1)v$ 就可变为 $dv/dt=(R_0(1-v)-1)v$,其解析解为 $v(t)=Kv_0/\{v_0-[v_0-K]e^{-rt}\}$,其中 $r=R_0-1$,$K=1-1/R_0$ 和 $v_0=T(0)/N$(初始被感染的用户比率)。于是,当 $t\to\infty$ 时,就有 $v(t)\to 1-1/R_0$,或等价地说,受害者人数 $T(t)=Nv(t)\to N(1-1/R_0)$。证毕。

上面定理 1 还隐含了另一个重要事实,由于最终受害者人数趋于 $N(1-1/R_0)=N-N/R_0=N-\gamma/\beta=N-$(康复率/传染率),即,如果康复型恶意代码不能被消灭,那么,最终受害者人数将基本上由比率“康复率/传染率”决定,或者说:如果传染率远远大于康复率,那么,基本上会全体受害;反之,受害者人数将维持在一个较小的数目之内。换句话说,对待康复型恶意代码,只要做好安全维护工作,那么,整体局面是可控的。

上面的定理 1 还告诉我们,只要控制住 R_0 使得 $R_0<1$,那么,就可成功地控制该恶意代码的爆发。由于 $R_0=(\beta N)/\gamma$,所以,我们可以增大康复率 γ(尽快恢复中毒终端),减少传染率 β(增加安全防护能力),减少初始人群数 N(隔离受害终端)等手段来控制 R_0。

1.3 疫型恶意代码的态势

有些恶意代码,当用户被为害后,只要康复了,那么,该用户就不会再被为害了,这类恶意代码就称为免疫型的。比如,利用系统漏洞的那些恶意代码,当受害用户,用现成的补丁程序,把相关漏洞补好后,该用户就不会再被伤害了,准确地说,不会再被同一个恶意代码伤害了。(人们对同一谣言,肯定会有免疫性的,只要辟谣者有相当的信任度。)

设 S、T、γ、β 和 N 等概念与 1.2 节相同,又记 $R(t)$ 为 t 时刻被康复(当然也就具有了免疫力)的用户数。于是,在任何一个时刻,都恒有 $N=S(t)+T(t)+R(t)$。为了使相关公式看起来简单一些(实质上是等价的),分别用 $S(t)/N$、$T(t)/N$ 和 $R(t)/N$ 去代替 $S(t)$、$T(t)$ 和 $R(t)$ 并且仍然采用原来的记号来表示 $S(t)$、$T(t)$ 和 $R(t)$,此时便有 $S(t)+T(t)+R(t)=1$,或者说,$S(t)$、$T(t)$ 和 $R(t)$ 分别代表暂未受害、正受害和受害康复且具有免疫力的用户,各占总用户数的比例。于是,仿照 1.2 节的分析,我们有如下 3 个微分方程:

$dS(t)/dt=-\beta TS$(这是因为 βT 是传染力,所以,被染人数的变化率就为 βTS)

$dT(t)/dt=\beta TS-\gamma T$(这是因为 γT 和 $\beta TS-\gamma T$ 分别为康复和受害数的变化率)

$dR(t)/dt=\gamma T$(康复人数变化率)

假定刚开始时,受害用户数为 $T(0)=T_0>0$;暂未受害的用户数为 $S(0)=S_0>0$;还没有用户具有免疫力,即 $R(0)=0$。

从第一个微分方程,有 $dS(t)/dt=-\beta TS<0$,所以,暂未受害的用户数始终随着时间 t 的增加而减少,并且以 0 为下限,故极限 $\lim\limits_{t\to\infty}S(t)=S_\infty$ 肯定存在(实际上,已经证明:$S_\infty=-\rho\text{Lambert } W\{-[\exp(-(T_0+S_0-\rho\ln S_0)/\rho)]/\rho\}$)。

从第二个微分方程 $dT(t)/dt=\beta TS-\gamma T=T(\beta S-\gamma)$,可以看出 $T(t)$ 的增减性依赖于 t 时刻 $S(t)$ 的大小。如果 $S_0<\gamma/\beta$,那么,$\beta S-\gamma<\beta S_0-\gamma<0$,于是,$dT(t)/dt<0$,即,当时间 $t\to\infty$ 时,有 $T_0>T(t)\to 0$,此时,恶意代码将被最终消灭。但是,如果 $S_0>\gamma/\beta=\rho$,那么,$T(t)$ 在某个时段内将会增加,从而导致恶意代码的危害呈爆发现象,但是,随着时间的进一步推移和 $S(t)$ 的递减,$T(t)$ 达到最大值后,又开始递减,并最终趋于 0。由此可知,免疫型恶意代码一定存在着临界现象,即,如果 $S_0>\rho$,则受害用户数爆增;否则,如果 $S_0<\rho$,则恶意代码处于可控状态;但是,无论是在哪种情况下,免疫型恶意代码将最终被消灭。提醒:读者别误会,此处意指的是,给定的某种免疫型恶意代码会最终消灭,但是,一旦产生新的免疫型恶意代码,那么,类似的上述

动力学过程又得重新演绎一次。

定义另一个参数 $F_0=\beta S_0/\gamma$,它刻画了一个受害者在平均染毒周期($1/\gamma$)内,所传染的人数,它也给出了该种恶意代码是否爆发的阈值。即,当 $F_0<1$ 时,此恶意代码不会爆发,并随着时间的推移,会自动消灭;当 $F_0>1$ 时,该恶意代码会在一定的时段内爆发,受害者人数达到一个最大值后,才开始递减,并最终消灭。$F_0<1$ 说明一个受害者在平均传染周期内,传染人数的个数小于 1,该恶意代码当然会自行消灭;$F_0>1$ 说明一个受害者在平均传染周期内传染的人数大于 1,故该恶意代码会在一定程度上爆发流行。归纳起来,我们有:

定理 2:针对免疫型恶意代码,当 $F_0<1$ 时,此恶意代码不会爆发,并随着时间的推移,会自动消灭;当 $F_0>1$ 时,该恶意代码会在一定的时段内爆发,受害者人数达到一个最大值 T_{max} 后,才开始递减,并最终消灭。更深入地,T_{max} 在总人数中所占的比例为 $1-\rho+\rho\ln(\rho/S_0)$,这里 S_0 表示刚开始时,暂未受害的人数比例。随着时间推移至无穷大,暂未受害和已有免疫力的人数的比例 S_∞ 和 R_∞ 分别为:

$S_\infty=-\rho\text{LambertW}\{-[\exp(-(T_0+S_0-\rho\ln S_0)/\rho)]/\rho\}$ 和 $R_\infty=N+\rho\text{LambertW}\{-[\exp(-(T_0+S_0-\rho\ln S_0)/\rho)]/\rho\}$

这里各相关参数的含义,见本小节中上面的描述。LambertW(.)是一种特殊的数学函数,见文献[15]中的第 12 章的 12.6 节。

本定理的前半部分,很形象,也是对安全界更有用的部分,而且已经在前面描述中给出了证明。本定理的后半部分其实也是生物医学中的已知结论,为避免陷入不必要的数学细节,我们在此略去。有特殊兴趣的读者可读文献[15]第 4 章的 4.3 节。

1.4　考虑开机和关机对免疫型恶意代码的影响

上面的所有分析,都假定活跃用户数固定为 N,但是,在实际情况下,当然有例外。比如,某用户主动关机后,任何恶意代码对他都不构成威胁,此时活跃用户就减少一个;当某用户终端中毒后被宕机,这里活跃用户数也减少一个;当新用户开机(或进入网络)后,他又可能成为恶意代码的攻击对象,这时,活跃用户数又增加一个等。

设 μ 是新用户开机率,g 是用户主动关机率,c 为被恶意代码攻击后的宕机概率,为简单计,都假定这些参数为常数。其它参数同上。于是,

在某个时刻,暂未受害的人数为 $S(t)$,那么,下一时刻,$S(t)$ 会因为用户主动关机而减少 $gS(t)$;会因为恶意代码的攻击而减少 βST;会因为新用户开机,而增加 $\mu N(t)$ 等。

在某个时刻,受害终端数为 $T(t)$,那么,下一时刻,$T(t)$ 会因为恶意代码的攻击而增加 βST;会因为宕机和主动关机而减少 $(g+c)T(t)$;会因为康复而减少 $\gamma T(t)$。

在某个时刻,已有免疫力的用户数为 $R(t)$,那么,下一时刻,$R(t)$ 会因为康复而增加 $\gamma T(t)$;会因为主动关机而减少 $gR(t)$。

因此,下面可以考虑两种特殊情况。

情况 1:假如没有宕机(即,$c=0$)并且主动关机率与新用户开机率相同(即,$\mu=g$),那么,各类终端数的变化情况,便可以用如下 3 个微分方程来描述:

$$dS(t)/dt=\mu N(t)-\beta T(t)S(t)-\mu S(t)$$
$$dT(t)/dt=\beta T(t)S(t)-\gamma T(t)-\mu T(t)$$
$$dR(t)/dt=\gamma T(t)-\mu R(t)$$

对这 3 个微分方程的解进行分析后(为突出重点,略去详细的数学过程。有特殊兴趣的读者,可以从生物数学的教材中找到答案,比如,文献[15]的第 4.4 节),我们可以仿照定理 2,得到如下的定理 3.1。

定理 3.1:记 $P_0=\beta/(\gamma+\mu)$,那么,在情况 1 之下,当 $P_0\leqslant 1$ 时,该免疫型恶意代码一定会随着时间的推移,最终自动消灭;当 $P_0\geqslant 1$ 时,该免疫型恶意代码一定会随着时间的推移,最终在 $S=1/P_0$、$T=\mu(P_0-1)/\beta$、$R=1-1/P_0-\mu(P_0-1)/\beta$ 点处达到全局渐近稳定,即,最终健康终端的比例为 $S=1/P_0$,受害终端的比例为 $T=\mu(P_0-1)/\beta$,获得免疫力的终端比例为 $R=1-1/P_0-\mu(P_0-1)/\beta$。

情况 2:有宕机发生(即 $c>0$),新用户开机率大于主动关机率(即 $\mu>g$)那么,各类人数的变化情况,便可以用如下 3 个微分方程来描述:

$$dS(t)/dt=\mu N(t)-\beta T(t)S(t)-gS(t)$$
$$dT(t)/dt=\beta T(t)S(t)-\gamma T(t)-cT(t)-gT(t)$$
$$dR(t)/dt=\gamma T(t)-gR(t)$$

这组微分方程的求解就不容易了!为了简化,我们把每台终端的平均开机率 $\mu N(t)$ 用一个常数 B 来代替,也初略地称其为开机率。于是,上面的 3 个微分方程就简化为:

$$dS(t)/dt=B-\beta T(t)S(t)-gS(t)$$
$$dT(t)/dt=\beta T(t)S(t)-\gamma T(t)-cT(t)-gT(t)$$
$$dR(t)/dt=\gamma T(t)-gR(t)$$

由于 $N(t)=S(t)+T(t)+R(t)$，所以，将这 3 个方程相加，又得到第 4 个方程：

$$dN(t)=B-cT(t)-gN(t)$$

与前面类似，分析这些微分方程的解后，可得到如下形象结果：

定理 3.2：记 $Q_0=\beta B/[g(\gamma+c+g)]$，那么，在情况 2 之下，当 $Q_0<1$ 时，该免疫型恶意代码一定会随着时间的推移，最终自动消灭；当 $Q_0>1$ 时，该免疫型恶意代码一定会随着时间的推移，最终在 $S^*=BgQ_0$、$T^*=(B-gS^*)/(\beta S^*)$、$N^*=(B-cT^*)/g$、$R^*=N^*-S^*-T^*$ 点处达到渐近稳定，即，最终的活跃用户数为 N^*，健康终端的比例为 S^*/N^*，受害终端的比例为 T^*/N^*，获得免疫力的终端比例为 $R^*=1-S^*/N^*-T^*/N^*$。

1.5 预防措施的效果分析

对付恶意代码其实都有许多预防措施的，但是，由于用户懒惰或不懂技术，总会有一些用户没采取预防措施。比如，针对利用某已知漏洞的恶意代码，厂商一般都会在病毒未大规模爆发前，发布相关的补丁程序，用户只要及时安装了这些补丁，那么，他的终端就已具备了免疫力；但事实是：一定有许多用户没打补丁。

我们虽然不能强求全体用户都采取预防措施，但是，如果有比例为 p 的用户采取了预防措施（比例为 q 的用户偷了懒，此处，$p+q=1$），那么，我们发现：只要当 p 足够大时，仍然能够消灭该恶意代码。

为简单计，我们在 1.4 节的情况 1 基础上，继续考虑问题。暂未受害的人中，有比例为 p 的终端直接获得了免疫力，于是，1.4 节的那 3 个微分方程就变成了：

$$dS(t)/dt=\mu qN(t)-\beta T(t)S(t)-\mu S(t)$$

$$dT(t)/dt=\beta T(t)S(t)-\gamma T(t)-\mu T(t)$$

$$dR(t)/dt=\mu pN(t)+\gamma T(t)-\mu R(t)$$

分析该方程组后（细节略去），我们有如下结论：

定理 4：在 1.4 节的情况 1 中，如果在暂未受害的终端中，采取了预防措施终端数的比例 $p\geqslant 1-1/P_0$，这里 $P_0=\beta/(\gamma+\mu)$，那么，该免疫型恶意代码一定会随着时间的推移，最终自动消灭。

此定理告诉我们：对付恶意代码，虽然不能指望全体人员都及时采取预防措施，但是，只要有足够多的人（占总人数比例超过 $P_0=\beta/(\gamma+\mu)$）重视安全，并及时采取了预防措施，那么，该恶意代码就一定是可控的，甚至会最终被消灭。

1.6 有潜伏期的恶意代码态势

许多恶意代码（比如，逻辑炸弹），在潜入受害终端后，并不立即作恶，而是等待时机成熟后再行动。假设仍然在 1.4 节的基础上来考虑问题，并且，暂未受害的终端，在变成受害终端前，先要经过一个潜伏阶段。用 $E(t)$ 表示 t 时刻已经中毒，但仍然处于潜伏期的终端数；用 η 表示潜伏终端变成受害终端的概率，于是，仿照 1.4 节，我们就有如下 4 个微分方程：

$$dS(t)/dt=\mu N(t)-\beta T(t)S(t)-\mu S(t)$$

$$dE(t)/dt=\beta T(t)S(t)-\mu E(t)-\eta E(t)$$

$$dT(t)/dt=\eta E(t)-\gamma T(t)-\mu T(t)$$

$$dR(t)/dt=\gamma T(t)-\mu R(t)$$

此微分方程组的分析方法与前面类似，我们就不再重复了。可见，用生物数学的成果去研究网络安全，其实是有很多事情能做的。

2 结束语

由于学科越分越细，肯定会出现这样的情况，即，不同学科的某些问题，其本质是一样的，于是，只要在一个学科中解决了相关问题，那么，另一个学科的同类问题也就迎刃而解了。

但事实却是：不同学科的人们，越来越封闭于自身学科，从而，做了许多不必要的重复性工作。在文献[11]中，我们已经发现，甚至像著名的《信息论》和《博弈论》，都可以融合成一套理论；此文又发现，原来传染病学中的许多东西也是可以用来研究计算机恶意代码的。我们相信生物学中一定还有其它成果和思路，可以借用来研究网络空间安全问题。但愿有更多的人，前来挖这个金矿。

在对付恶意代码方面，过去安全专家主要聚焦于如何从微观上战胜它，这就像医生研究某种具体的药物来医治传染病一样。但是，经过几百年来的实践，医生们已经发现：针对传染病，药物治疗重要，但是，更重要的是，要根据传染病的传播特征，从宏观上控制传染病。比如，若无隔离，那么，药物再好，也不能控制 SARS 的爆发。

因此，现在已经到了安全专家该向医生学习的时候了，我们必须刷新自己的观念。记住：从微观上对付恶意代码个案（比如，研发漏洞补丁）虽然重要，但是，从宏观上控制恶意代码流行爆发更重要，只有搞清楚恶意代码扩散的动力学特性后，才能够稳准狠地对付

流行趋势。本文在这方面只是一个开始,前途很光明,任务也很艰巨。

希望安全专家们前往生物领域,挖掘医生们防流感的更多法宝;更希望有医生(特别是生物数学家)前来增援网络空间安全领域的相关研究。

最后,再重复说明一下:恶意代码并非都以"病毒式"传播,比如,有些恶意代码只针对特定目标,它肯定以隐藏为主,不会再传播出去攻击别人。所以,本文的宏观行为对非病毒式恶意代码是无效的。

参考文献:

[1] 杨义先,钮心忻. 安全通论(1)之"经络篇"[EB/OL]. http://blog.sciencenet.cn/blog-453322-944217.html,2015-12-18.

[2] 杨义先,钮心忻. 安全通论(2):攻防篇之"盲对抗"[EB/OL]. http://blog.sciencenet.cn/blog-453322-947304.html,2016-01-01.

[3] 杨义先,钮心忻. 安全通论(3):攻防篇之"非盲对抗"之"石头剪刀布游戏"[EB/OL]. http://blog.sciencenet.cn/blog-453322-948089.html,2016-01-04.

[4] 杨义先,钮心忻. 安全通论(4):攻防篇之"非盲对抗"之"童趣游戏"[EB/OL]. http://blog.sciencenet.cn/blog-453322-949155.html,2016-01-09.

[5] 杨义先,钮心忻. 安全通论(5):攻防篇之"非盲对抗"收官作及"劝酒令"[EB/OL]. http://blog.sciencenet.cn/blog-453322-950146.html,2016-01-13.

[6] 杨义先,钮心忻. 安全通论(6):攻防篇之"多人盲对抗"[EB/OL]. http://blog.sciencenet.cn/blog-453322-954445.html,2016-02-04.

[7] 杨义先,钮心忻. 安全通论(7):黑客篇之"战术研究"[EB/OL]. http://blog.sciencenet.cn/blog-453322-956051.html,2016-02-14.

[8] 杨义先,钮心忻. 安全通论(8):黑客篇之"战略研究"[EB/OL]. http://blog.sciencenet.cn/blog-453322-958609.html,2016-02-25.

[9] 杨义先,钮心忻. 安全通论(9):红客篇[EB/OL]. http://blog.sciencenet.cn/blog-453322-960372.html,2016-03-04.

[10] 杨义先,钮心忻. 安全通论(10):攻防一体的输赢次数极限[EB/OL]. http://blog.sciencenet.cn/blog-453322-984644.html,2016-06-14.

[11] 杨义先,钮心忻. 安全通论(11):信息论、博弈论与安全通论的融合[EB/OL]. http://blog.sciencenet.cn/blog-453322-989745.html,2016-07-11.

[12] 杨义先,钮心忻. 安全通论(12):对话的数学理论[EB/OL]. http://blog.sciencenet.cn/blog-453322-993540.html,2016-07-30.

[13] 杨义先,钮心忻. 安全通论(13):沙盘演练的最佳攻防对策计算[EB/OL]. http://blog.sciencenet.cn/blog-453322-1000428.html,2016-09-02.

[14] 杨义先. 刷新你的安全观念[EB/OL]. http://blog.sciencenet.cn/blog-453322-983276.html,2016-06-08.

[15] 肖燕妮,周义仓,唐三一. 生物数学原理[M]. 西安:西安交通大学出版社,2012.