一种基于生长理论的系统漏洞发现预测模型

唐成华[1,2]　潘然[1]　李海东[3]　周江山[2]　强保华[1,3]

（桂林电子科技大学 广西可信软件重点实验室　广西 桂林 541004）[1]
（广西密码学与信息安全重点实验室　广西 桂林 541004）[2]
（广西云计算与大数据协同创新中心　广西 桂林 541004）[3]

摘　要　系统漏洞是造成网络空间安全威胁的主要内在原因。针对系统漏洞的有效发现及预测问题，提出了一种基于生长理论的系统漏洞发现预测模型。首先分析漏洞发现规律，引入生长曲线的概念，确定了漏洞发现增长的阶段特征；其次在生长理论周期表达基础上，描述系统漏洞发现过程与时间的关系，提出系统漏洞发现的预测过程，以及改进后的 PMGTV 模型；最后在实验中与其他模型进行了对比和有效性等分析，在误差平方和 SSE 以及卡方值χ^2方面表现最好。结果表明，该模型在对系统漏洞发现的预测方面更有准确，为采取有效安全策略、提高软件质量等方面提供了一种可靠依据。

关键词　系统漏洞，漏洞发现，漏洞预测，生长曲线，网络安全

中图法分类号：TP301　　**文献标识码**：A

Predicting system vulnerability discovery with growth theory

TANG Cheng-hua[1,2]　PAN Ran[1]　LI Hai-dong[3]　ZHOU Jiang-shan[2]　QIANG Bao-hua[1,3]

(Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China)[1]
(Guangxi Key Laboratory of Cryptography and Information Security, Guilin 541004, China)[2]
(Guangxi Cloud Computing and Big Data Collaborative Innovation Center, Guilin 541004, China)[3]

Abstract　System vulnerabilities are the main internal causes of cyberspace security threats. Aiming at the effective discovery and prediction of system vulnerabilities, a system vulnerability detection and prediction model based on growth theory is proposed. Firstly, the rule of vulnerability discovery is analyzed, and the concept of growth curve is introduced to determine the stage characteristics of vulnerability discovery growth. Secondly, based on the periodic expression of growth theory, the relationship between system vulnerability discovery process and time is described, and the prediction process of system vulnerability discovery and the improved PMGTV model are proposed. Finally, it is compared with other models in the experiment and the validity is analyzed. It performs best in the sum of squares for error (SSE) and the Chi-square value χ^2. Results show that the model is more accurate in the prediction of system vulnerability discovery, and provides a reliable basis for taking effective security strategies and improving software quality.

Key words　System vulnerability, Vulnerability discovery, Vulnerability prediction, Growth curve, Network security

系统漏洞是指应用软件或操作系统软件在逻辑设计上的缺陷或在编写时产生的错误，这种缺陷或错误能够被不法者利用，从而引起代码异常、信息泄露、系统瘫痪等[1]。系统漏洞对网络空间安全的威胁越来越大，是造成系统故障的主要内在原因，也是系统安全评估的重要指标之一。因此，深入研究并发现系统漏洞，尤其是在漏洞发现前进行有效预测并采取适当安全策略，对于提高软件质量、减少系统故障损失具有重要的实践意义[2]。

由于漏洞是软件系统内部的一种固有缺陷，所以其数量是有限的。从系统设计开发完成到投入使用，漏洞的发现从零开始增长，在此过程中，因为不同应用阶段的情况、新漏洞的发现和旧漏洞的修补等，漏洞发现呈现不同的增长速度，其内部的未知漏洞数量会越来越少，总漏洞的发现也趋向于一固定值，系统也不断发展完善并趋于成熟。

现有的漏洞检测技术无法检测出所有的软件系统的漏洞。着眼于从宏观角度分析软件的漏洞数量，学术界从不同角度探讨漏洞数量的估计与预测，主要包括基于软件发布周期（时间）的预测和基于软件特征的预测。在基于发布周期方面，Anderson 最早将类比热动力学 AT（Anderson Thermodynamic Model）模型引入研究软件可靠性及漏洞发现[3]；另外还有 Musa 提出对数泊松 LP（Logarithmic Poisson Model）模型[4]，认为漏洞数量和软件发现时间之间是一种对数关系；Rescorla 则提出了某些情况下漏洞增长过程是一个指数增长的 RE（Rescorla Exponential Model）模型[5]；Kim 针对部分多版本开源软件系统间共享源代码度量关系，建立一种类线性增长的漏洞发现模型，找出了软件的演进与漏洞检测之间的关系[6]；Browne 不考虑软件本身属性特征下，根据 CERT 报告即得出漏洞总数与时间的平方成正比的结论[7]。以上这些模型或观点并不具备一般性，实验数据误差较大，无法正确分析软件系统漏洞情况。Alhazmi 等人对多种漏洞

本文受国家自然科学基金 (61462020, 61762025)；广西可信软件重点实验室基金(kx201506)；广西密码学与信息安全重点实验室基金(GCIS201619)；广西云计算与大数据协同创新中心项目(YF17101)；广西高等学校高水平创新团队及卓越学者计划资助。

唐成华（1974-），男，博士后，副教授，硕士生导师，CCF会员(E200037781M)，主要研究方向为网络信息安全、智能信息处理，Email：tch@guet.edu.cn；潘然（1992-），男，硕士生，主要研究方向为系统漏洞检测与分析；李海东（1992-），男，硕士生，主要研究方向为网络信息存储安全；周江山（1995-），男，硕士生，主要研究方向为网络信息安全；强保华（1972-），男，博士后，教授，主要研究方向为智能信息处理。

类型进行了研究，提出了软件漏洞度量、发布和预测等一系列的 AML（Alhazmi-Malaiya Logistic Model）逻辑模型[8]，认为漏洞增长具有平缓期、快速期和停滞期三阶段，该模型体现了软件漏洞发现数量在时间上的一般符合，具有广泛的适用性。在基于软件特征方面，主要借采用件缺陷预测模型的思路和方法，根据软件代码度量的复杂性、耦合性和内聚性等指标建立训练模型[9-10]，进而预测新代码中的漏洞存在的可能性及其数量。Yang 提出了软件代码数量的规模与软件缺陷之间的关系[11]；Wei 提出了基于软件代码组件之间依赖图关系的漏洞预测方法[12]；Shin 深入分析代码之间调用关系并获取其与软件缺陷的可能性关系[13]。这类以软件特征为分析对象的预测方法，关键在于获取与软件缺陷有关的度量数据，进而对程序模块的缺陷倾向性、缺陷数量进行预测，属于静态软件漏洞预测的范畴，这里暂不考虑。本文借鉴 AML 模型的思想，并引入生长理论来描述系统漏洞发现过程与时间的关系，建立漏洞预测模型。实验结果表明模型具有漏洞预测有效性，且在预测中较现有模型更为灵活。

1 系统软件漏洞发现的规律

系统漏洞的发现通常伴随软件整个生命周期，在系统软件尚未开发完成时即存在，与代码规模和内在复杂度有关。但在开发和测试阶段亦是存在，影响此部分漏洞发现的因素与代码修改、开发人员经验、模块依赖性和项目团队组织架构有很大关系[14]，对此部分不做研究。本文探讨漏洞在系统软件发布上市后的变化规律。一个系统软件漏洞的发现速度并不是保持一个不变的数值，系统软件漏洞的发现速度也有一定规律可循。系统软件发布上市后影响漏洞发现速度的因素有很多，包括用户规模、黑客攻击等。软件的使用者越多，软件中存在的漏洞就越容易被发现，而黑客也往往会挖掘具有大规模用户的系统中的漏洞并攻击。随着系统软件的不断使用和完善，其未发现漏洞会越来越少，漏洞的发现速度也随之降低。

选取系统软件 win_2000、win_xp（数据来源于中国国家漏洞数据库）统计出它们的漏洞发现速度如图 1 所示。

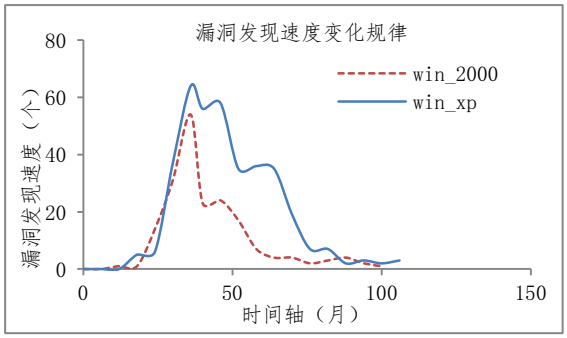

图 1　win_2000 和 win_xp 漏洞发现速度随时间的变化

根据图 1 可以看出，系统软件漏洞的发现速度可以分为三个阶段：在软件发布上市初期，漏洞增长较为缓慢；随后系统软件进入市场一段时间后，其漏洞的发现速度开始增长；但增长不会一直持续下去，再接着漏洞的发现速度开始呈现下降趋势；最后漏洞发现速度变的平稳，几乎不再增加。对于生长理论来说，事物总是经过发生、发展、成熟三个阶段，而每一个阶段的发展速度各不相同。通常在发生阶段，变化速度较为缓慢；在发展阶段，变化速度加快；在成熟阶段，变化速度又趋缓慢，按上述三个阶段发展规律得到的变化曲线称为生长曲线[15]。这种生长理论的曲线规律与漏洞发现速度呈现一致性。

2 漏洞预测模型

经典的 Logistic 方程是一种生态学生长理论方程[16]，其根据生物学原理作出某种假设前提并建立相关微分方程和模型，然后代入初始条件和临界条件求解出生长过程。模型过程具有逻辑性强、实用范围广、参数可解释等特点，作为一条标准的 S 型曲线，在自然界和社会中很多事物的变化体现了这一规律，因此在生态学和经济学中已得到广泛应用。Logistic 方程是根据生长理论可知生物生长有一定的上限，类比漏洞发现的增长过程来说，系统软件漏洞数量最终趋向一个饱和值。

设在 t 时刻，漏洞的发现数量为 y。因为漏洞发现的增长速度不是一个定值，漏洞发现增长速度与当前漏洞发现的规模有关，结合生长理论，漏洞发现增长速度是关于 y 的函数，即有如下式：

$$\frac{dy}{dt} = f(y) \tag{1}$$

用泰勒级数表示，即有：

$$f(y) = C_0 + C_1 y + C_2 y^2 + \cdots\cdots + C_{n-1} y^{n-1} + C_n y^n \tag{2}$$

其中 C_0、C_1、…、C_n 为常数。在软件系统发布初期（此时系统漏洞发现数量为 0）漏洞增长速度接近于 0；在系统软件成熟末期，漏洞发现数量趋于一个饱和值（设为 M），漏洞增长速度也接近 0。据此有如下关系式：

$$\begin{cases} \frac{dy}{dt} = f(0) = 0 \\ \frac{dy}{dt} = f(M) = 0 \end{cases} \tag{3}$$

将$f(0) = 0$, 代入式(2)中可得 $C_0 = 0$。当漏洞数量趋近饱和值 M 时，$f(M) = 0$, 代入式(2)中，可知$f(y) = 0$有两个根，当级数在y^2 停止时有$f(y)$的最简形式：

$$\frac{dy}{dt} = C_1 y + C_2 y^2 \tag{4}$$

将 y=M 代入式(4)中，得$C_1 = -C_2 M$，再将C_2用C_1替换可得出基于生长理论的漏洞预测模型的微分形式：

$$\frac{dy}{dt} = C_1 y(1 - \frac{y}{M}) \tag{5}$$

对微分形式(5)进行求解可得如下：

$$y = \frac{M}{1+e^{(a-C_1 t)}} \tag{6}$$

其中a是求解过程中引入的常量参数，式(6)是一个 S 型曲线模型。

通过对软件系统漏洞发现速度随软件市场周期的变化规律的分析，生长曲线这一特点与系统漏洞发现的过程相似，基于生长曲线来描述软件系统漏洞发现的过程。选取 win_server_2003 和 win_xp 两款系统软件，采用式(6)分析和描述系统漏洞累计数量与软件发布的市场周期的关系，如图

2 所示。

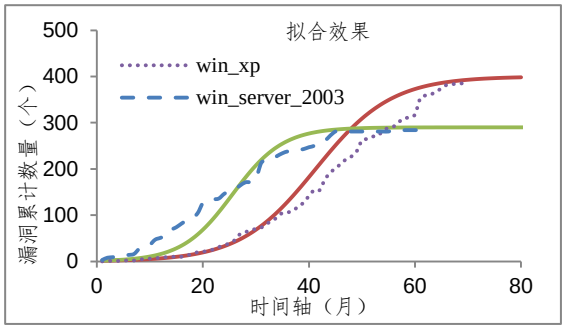

图 2　基于生长曲线的漏洞发现的拟合

图 2 反映出的漏洞发现拟合效果并不理想。因为预测模型式(6)是一个标准的 S 型曲线，漏洞发现的增长实际过程并非完全符合其规律和形式。作为研究生物生长趋势而提出的生长曲线模型，常应用于植物生长模拟和预测。这种植物的自然界生长环境相对来说并不复杂。但对于软件系统漏洞而言，影响漏洞增长速度的有诸如用户、语言、工具、团队、攻击等因素，对软件漏洞的出现速度都有很大影响。标准 S 生长曲线模型在模拟软件漏洞增长时会有很大误差。

为了增加模型的灵活性和适用性，对式(6)进行改进。为了便于区分将公式(6)中的$e^{(a-C_1t)}$替换成$e^{A(B-t)}$，用$S(t)$替换y。并对预测模型(6)的分母进行修改：

$$S(t) = \frac{M}{[1+e^{A(B-t)}]^{\alpha^t+\beta}} \quad 0<\alpha<1, \beta>0 \qquad (7)$$

其中 A、B、M、α 和 β 均为常数，对常数 α 和 β 限定范围，使其更适合于对漏洞的预测。将式(7)命名为 PMGTV 模型。

PMGTV 模型主要具有以下特性：

（1）模型曲线分为三个阶段：初期，漏洞增长缓慢；中期，漏洞增长较为迅速；末期，漏洞增长缓慢并趋于稳定。

（2）递增性：模型曲线$S(t)$是关于 t 的单调递增函数，即软件漏洞累计数量只会增多不会减少。

递增性证明：

$$\because S'(t) = \left\{\frac{M}{[1+e^{A(B-t)}]^{\alpha^t+\beta}}\right\}' - \frac{M\left\{e^{(\alpha^t+\beta)\ln[1+e^{A(B-t)}]}\right\}'}{\{[1+e^{A(B-t)}]^{\alpha^t+\beta}\}^2}$$

$$= -\frac{Me^{(\alpha^t+\beta)\ln[1+e^{A(B-t)}]}\{(\alpha^t+\beta)\ln[1+e^{A(B-t)}]\}'}{\{[1+e^{A(B-t)}]^{\alpha^t+\beta}\}^2}$$

$$= -\left\{Me^{(\alpha^t+\beta)\ln[1+e^{A(B-t)}]}\right\}\left\{\alpha^t \ln\alpha \ln[1+e^{A(B-t)}] - \frac{(\alpha^t+\beta)Ae^{A(B-t)}}{1+e^{A(B-t)}}\right\}/\{[1+e^{A(B-t)}]^{\alpha^t+\beta}\}^2$$

$$= \left\{Me^{(\alpha^t+\beta)\ln[1+e^{A(B-t)}]}\right\}\left\{\frac{(\alpha^t+\beta)Ae^{A(B-t)}}{1+e^{A(B-t)}} + \alpha^t(-\ln\alpha)\ln[1+e^{A(B-t)}]\right\}$$

$$/\{[1+e^{A(B-t)}]^{\alpha^t+\beta}\}^2 \xrightarrow{0<\alpha<1\to(-\ln\alpha)>0} S'(t) > 0$$

∴ 曲线$S(t)$是关于 t 单调递增的。

（3）具有上下渐进线：当$t\to\infty$时，模型曲线$S(t)$有上渐近线，$S(t)\to M$，即软件漏洞发现数量随着时间的推移会趋向一个饱和值。同时当$t\to 0$时$S(t)$有下渐近线：

$$t\to 0,\ S(t)\to \frac{M}{[1+e^{AB}]^{1+\beta}}$$

渐近线证明：

$$\because lim_{t\to 0}S(t) = lim_{t\to 0}\frac{M}{[1+e^{A(B-t)}]^{\alpha^t+\beta}} = \frac{M}{[1+e^{AB}]^{1+\beta}}\text{（常量）}$$

$$\lim_{t\to\infty}S(t) = \lim_{t\to\infty}\frac{M}{[1+e^{A(B-t)}]^{\alpha^t+\beta}}$$

$$= \frac{M}{\lim_{t\to\infty}[1+e^{A(B-t)}]^{\alpha^t+\beta}} = \frac{M}{e^{\lim_{t\to\infty}(\alpha^t+\beta)\ln[1+e^{A(B-t)}]}}$$

$$\xRightarrow{\alpha<1} \lim_{t\to\infty}S(t) = \frac{M}{e^0} = M\text{（上渐进线）}$$

∴ 曲线S(t)具有上下渐进线。

3 实验分析及评价

3.1　有效性实验

为验证模型的有效性，从中国国家信息安全漏洞库选取四款系统软件，然后用 PMGTV 模型来模拟漏洞的增长过程，首先给定模型各参数取值（精确到小数点后面 4 位），如表 1 所示。

表 1 PMGTV 模型拟合参数值

	M	A	B	α	β
win_xp	398.8761	0.1193	48.9811	0.9013	0.6712
win_server_2003	288.8254	0.0983	25.9221	0.7143	0.2101
mac_os_server	374.8106	0.1572	29.9244	0.5003	0.3706
unbuntu_linux	251.6877	0.2114	39.8808	0.8919	0.7117

系统软件漏洞的增长过程如图 3 所示。

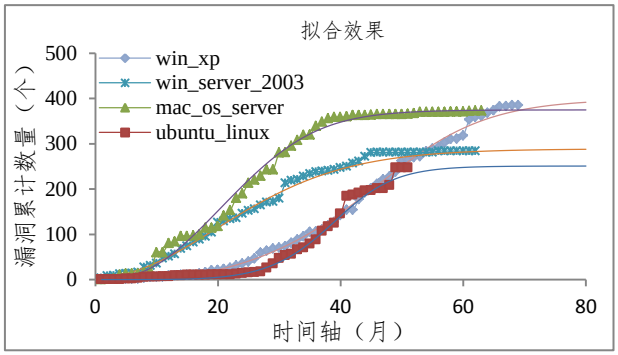

图 3　基于改进生长曲线的漏洞的增长拟合

由图 3 可以看出 PMGTV 对 win_xp、win_server_2003、mac_os_server、ubuntu_linux 这四款系统软件的漏洞增长过程的预测效果良好。其中对 mac_os_server 的拟合前期误差较大，但在后期的拟合效果中表现良好；对 win_server_2003 的拟合中前期拟合效果很好，中后期拟合实际的漏洞走势波动较大，但整体能够拟合；对 win_xp 的拟合效果是这四个系统漏洞中拟合最好的一个；对 ubuntu_linux 的拟合前中期都表现良好，后期有一些小误差，但整体亦是有效拟合。因此 PMGTV 模型在对漏洞的预测中是有效的。

3.2 模型评价

使用误差平方和（SSE）以及卡方值χ^2来作为评价模型的指标依据，与现有的 AT、AML、RE 和 LP 模型进行漏洞预测拟合上的误差性比较。

SSE 和χ^2计算公式如下：

$$\mathrm{SSE}=\sum_{i=1}^{n}(Y_{actual}-Y_{predict})^2 \tag{8}$$

$$\chi^2=\sum_{i=1}^{n}\frac{(f_i-p_i)^2}{p_i} \tag{9}$$

式中Y_{actual}、$Y_{predict}$分别表示实际值和预测值在同样的数据集下，SSE 值越小，误差越小，模型拟合效果越好。f_i、p_i 分别表示实际值和理论值，χ^2值越小，拟合效果越好，反映出模型具有更好的漏洞发现效果。

几种模型的描述形式如表 2 所示。

表 2 几种模型的描述形式

	模型公式	模型特征
AT	$S(t)=(k/\gamma)\ln(Ct)$	对数模型
AML	$S(t)=\mathrm{B}/(BCe^{-ABt}+1)$	标准 S 型曲线
RE	$S(t)=N(1-e^{-\gamma t})$	指数模型
LP	$S(t)=\beta_0\ln(1+\beta_1 t)$	泊松对数模型
PMGTV	$S(t)=M/([1+e^{A(B-t)}]^{\alpha t+\beta})\ \ 0<\alpha<1,\beta>0$	非标准 S 型曲线

从表 2 可知，AT 模型和 LP 模型较为相似它们都是对数模型，两者在预测性能上也较为接近；RE 为指数模型，实验中发现其预测性能较差；AML 模型是标准的 S 型模型，与本文提出的 PMGTV 模型类似，但 PMGTV 模型更为灵活，它们在拟合软件系统漏洞发现时的卡方值和误差平方和值的性能表现如表 3 所示。

表 3 性能对比（“*”表示拟合误差过大不能拟合）

	AT		AML		RE		LP		PMGTV	
	SSE	χ^2	SSE	χ^2	SSE	χ^2	SSE	χ^2	SSE	χ^2
win_xp	*	*	3475.584	98.308	*	*	*	*	3846.014	88.474
win_server_2003	*	*	7658.477	49.349	*	*	9806.005	249.069	4001.099	22.149
mac_os_server	*	*	12493.258	210.354	19839.147	709.226	17147.428	511.514	9706.082	119.004
unbuntu_linux	*	*	4873.221	436.254	*	*	*	*	4050.813	264.251

从表 3 可以看出，AT 模型并不能拟合这几款软件，RE 模型能够拟合 mac_os_server 的漏洞随时间的变化过程，但是 SSE 和χ^2值很大，因此有较大的误差。LP 模型能够拟合 win_server_2003 和 mac_os_server 系统软件漏洞随时间的变化过程，但 SSE 和χ^2值也处于较高的值故而有较大的误差。AML 模型可以拟合这 4 种系统软件漏洞随时间的变化过程，其中在对 win_xp 拟合时，误差与 PMGTV 模型的误差接近，但在对其他系统软件的拟合时，误差明显大于 PMGTV 模型。

从拟合效果的分析可知，对数模型（AT、LP）和指数模型（RE）在拟合效果上表现极差，并不能准确描述漏洞发现数量随时间的变化规律，而 AML 和 PMGTV 模型拟性能良好，这也说明了生长理论描述漏洞发现过程的有效性。

3.3 模型预测效果

选取 win_2000 和 win_7 系统软件来验证模型的预测效果。其中 win_2000 发布于 2000 年 2 月。以 win_2000 系统软件 2000 年-2010 年的漏洞数据作为样本数据预测其 2011 年-2018 年的漏洞增长情况。根据 win_2000 发布的前 10 年的漏洞数据拟合出的模型如下：

$$S(t)=\frac{187.5973}{[1+e^{0.2708(38.3197-t)}]^{0.4619t+0.5113}} \tag{8}$$

根据模型相关的参数，预测最终漏洞数量趋近于饱和值 187.5973 左右，实际查询漏洞数据库后的截止目前为止漏洞累计数量为 181 个。

win_7 发布时间为 2009 年 10 月，根据经验其处于漏洞增长期，以 2009 年发布开始截止到目前（2018 年）的漏洞数据作为参考预测未来（2018 年以后）的漏洞走势，拟合出的模型如下：

$$S(t)=\frac{983.6903}{[1+e^{0.0911(51.3347-t)}]^{0.8906t+0.7714}} \tag{9}$$

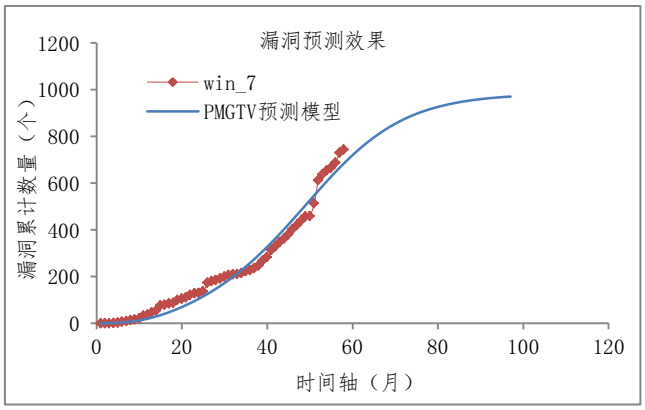

图 4 win_7 漏洞拟合预测

如图 4 所示，实线表示的数据是 win_7 截止目前的真实漏洞数据，虚线表示的数据是模型对 win_7 的预测数据。根据预测模型的参数和拟合图像可以预测 win_7 在未来漏洞的变化趋势，根据模型图可知 win_7 系统正处于漏洞的增长期，随后其会进入漏洞增长的平缓期，可以预测，最终其漏洞发现数量会趋近一个饱和值，该值大约为 983.6903。

结束语 目前在基于系统发布周期的软件漏洞发现检测与预测方面，传统的 AT 模型、LP 模型、RE 模型等各有自己的优点，但不具备一般性。根据软件漏洞增长过程具有类似生长规律这一特点，借鉴 AML 模型的思想，建立基于生长理论的漏洞发现模型 PMGTV，描述了系统漏洞发现过程与时间的关系，并对模型进行了有效性验证和与其他模型的对比分析。实验结果表明，PMGTV 模型在预测漏洞发现过程方面更为准确和灵活，能有效地帮助解决提高软件质量、减少系统故障损失等问题。在系统软件生命周期中，总是会在某几段时间内 PMGTV 模型难以拟合且误差较大，在这些时间段内系统一般都是进行了更新或升级，系统升级总是会引入新的代码，同时也极有可能引入新的漏洞。在未来的工作中，我们将会进一步探究系统更新或者升级对后续漏洞发现规律的影响，并完善本文模型继续提高其预测准确度。

参 考 文 献

[1] CHEN Kai, FENG Dengguo, SU Purui, et al. Multi-cycle vulnerability discovery model for prediction[J]. Journal of Software, 2010,21(9): 2367-2375.
陈恺,冯登国,苏璞睿,等. 一种多周期漏洞发布预测模型[J]. 软件学报, 2010,21(9):2367-2375.

[2] Nie Chujiang, Zhao Xianfeng, Chen Kai, et al. An software vulnerability number prediction model based on micro-parameters[J]. Journal of Computer Research and Development, 2011,48(7):1279-1287.
聂楚江,赵险峰,陈恺,等. 一种微观漏洞数量预测模型[J]. 计算机研究与发展, 2011,48(7):1279-1287.

[3] Anderson R. Security in open versus closed systems-the dance of Boltzmann, Coase and Moore[C]// Proceedings of the Conference on Open Source Software Economics. Cambridge, UK: MIT Press, 2002:1-15.

[4] Musa J D, Iannino A, Okumoto K. Software reliability engineering[M]. NY: McGraw-Hill, 1999:193-223.

[5] Rescorla E. Is finding security holes a good idea[J]. IEEE Security and Privacy, 2005,3(1):14-19.

[6] Kim J, Malaiya Y K, Ray I. Vulnerability discovery in multi-version software systems[C]// Proceedings of IEEE International Symposium on High Assurance Systems Engineering, USA: IEEE Computer Society, 2007:141-148.

[7] Browne H K, Arbaugh W A, McHugh J, et al. A trend analysis of exploitations[C]// Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy, Oakland: IEEE Press, 2001:214-219.

[8] Woo S W, Joh H, Alhazmi O H, et al. Modeling vulnerability discovery process in Apache and IIS HTTP servers[J]. Computers and Security, 2011,30(1):50-62.

[9] Li Zhiqiang, Jing Xiaoyuan, Zhu Xiaoke. Progress on approaches to software defect prediction[J]. IET Software, 2018,12(3):161-175.

[10] Stuchman J, Walden J, Scandariato R, et al. The effect of dimensionality reduction on software vulnerability prediction models[J]. IEEE Transactions on Reliability, 2017, 66(1):17-37.

[11] Yang Yibiao, Zhou Yuming, Lu Hongmin, et al. Are slice-based cohesion metrics actually useful in effort-aware post-release fault-proneness prediction? an empirical study[J]. IEEE Trans. on Software Engineering, 2015,41(4): 331-357.

[12] Wei Shengjin, He Tao, Hu Changzhen, et al. Predicting software security vulnerabilities with component dependency graphs[J]. Transactions of Beijing Institute of Technology, 2018,38(5):525-530.
危胜军,何涛,胡昌振,等. 基于组件依赖图的软件安全漏洞预测方法[J]. 北京理工大学学报, 2018,38(5):525-530.

[13] Shin Y, Bell R M, Ostrand T J, et al. On the use of calling structure information to improve fault prediction[J]. Empirical Software Engineering, 2012,17(4-5):390-423.

[14] CHEN Xiang, GU Qing, LIU Wangshu, et al. Survey of Static Software Defect Prediction[J]. Journal of Software, 2016,27(1):1-25.
陈翔,顾庆,刘望舒,等. 静态软件缺陷预测方法研究[J]. 软件学报, 2016,27(1):1-25.

[15] Rana R, Staron M, Berger C, et al. Selecting software reliability growth models and improving their predictive accuracy using historical projects data[J]. Journal of Systems and Software, 2014,98:59-78.

[16] Jukka R, Sami H, Ville L. The sigmoidal growth of operating system security vulnerabilities: an empirical revisit[J]. Computers and Security, 2015,55:1-20.