ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.21, No.9, September 2010, pp.2367−2375
doi: 10.3724/SP.J.1001.2010.03626

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

一种多周期漏洞发布预测模型*

陈 恺[1,2,3+], 冯登国[1,2], 苏璞睿[2], 聂楚江[2], 张晓菲[4]

[1](中国科学院 研究生院 信息安全国家重点实验室,北京 100049)

[2](中国科学院 软件研究所 信息安全国家重点实验室,北京 100190)

[3](信息安全共性技术国家工程研究中心,北京 100190)

[4](中国信息安全产品测评认证中心,北京 100089)

Multi-Cycle Vulnerability Discovery Model for Prediction

CHEN Kai[1,2,3+], FENG Deng-Guo[1,2], SU Pu-Rui[2], NIE Chu-Jiang[2], ZHANG Xiao-Fei[4]

[1](State Key Laboratory of Information Security, Graduate University, The Chinese Academy of Sciences, Beijing 100049, China)

[2](State Key Laboratory of Information Security, Institute of Software, The Chinese Academy of Sciences, Beijing 100190, China)

[3](National Engineering Research Center for Information Security, Beijing 100190, China)

[4](China Information Technology Security Certification Center, Beijing 100089, China)

+ Corresponding author: E-mail: chenk@is.iscas.ac.cn

Chen K, Feng DG, Su PE, Nie CJ, Zhang XF. Multi-Cycle vulnerability discovery model for prediction. *Journal of Software*, 2010,21(9):2367−2375. http://www.jos.org.cn/1000-9825/3626.htm

Abstract: This paper presents a multi-cycle vulnerability discovery model which shows the vulnerability discovery process and the relationship between the number of vulnerabilities and their release time. The model makes use of a cycle, which expands the scope of old models. A method is proposed to compute the parameters of this model to fit the discover process of the target software. Different rules are also given to find the initial values. Experiments are made on eight Windows operating systems. The results show that this model is more effective and more accurate than current models.

Key words: vulnerability; predict; model; multi-cycle; CSF

摘 要: 提出了一种多周期漏洞发布预测模型,描述了漏洞发现数量与时间的关系,预测漏洞发布过程.该模型引入周期概念,扩展了原单一增长过程的漏洞发布预测模型,增大了现有模型的适用范围.提出了相关参数的计算方法与初值选取方法,对8个版本的Windows操作系统进行实验分析.结果表明,该模型增加了预测过程的有效性,同时提高了预测结果的准确性.

关键词: 漏洞;预测;模型;多周期;CSF

中图法分类号: TP309 **文献标识码**: A

* Supported by the National Natural Science Foundation of China under Grant Nos.60703076, 60970028 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant Nos.2006AA01Z412, 2007AA01Z451, 2007AA01Z475, 2007AA01Z465, 2007AA01A414 (国家高技术研究发展计划(863))

Received 2008-09-03; Accepted 2009-04-10

软件安全漏洞是软件本身的缺陷,它可能引起代码异常、系统瘫痪,甚至被黑客利用进行入侵,引起更大的损失.因此,软件漏洞数量是软件安全性评估的重要指标之一,也是信息系统安全测评考量的重要因素之一.漏洞预测与漏洞分析不同:漏洞分析重点在于精确检测出软件漏洞的位置并加以修补,而漏洞预测着重于从宏观角度分析软件的漏洞数量.随着软件漏洞逐一被发现,对软件剩下的未知漏洞进行检测所花费的投入呈指数级增长,不可能通过漏洞分析的方法准确确定软件漏洞的数量,因此产生了从宏观角度的进行漏洞预测的方法,奠定了软件安全风险量化的可行性[1].

长久以来,研究人员从不同角度进行了软件漏洞总数与发布周期的预测.漏洞总数预测即预测软件在其生命周期的某个阶段内存在的漏洞数量,Akiyama,Halstead 等人在此做了一定工作,提出了软件漏洞与代码体积之间的关系[2].Takahashi[3]和 Alhazmi[4]分别提出了漏洞密度的概念,继而,人们根据软件生命周期中的不同阶段对漏洞密度进行估算.此外,还有基于软件复杂度的漏洞预测技术与基于多维软件度量元的缺陷预测技术[2].这类方法多用于软件工程中某个阶段的预测,且由于不同开发商的开发过程和开发质量的差异,这类方法存在较大的误差.即使是同一开发商的同一软件,由于软件本身不同及模块之间复杂度的差异,漏洞平均分布于软件中这一基础条件被破坏,使得估计结果不准确.同时,这类方式仅能估算最终的漏洞总数,并不能对漏洞发布过程中的时间特性进行预测.

为了能对漏洞发布周期进行建模,人们提出了漏洞发布时间预测模型.Rescorla[5]提出了线性模型和指数模型,但是这两种模型均不具有显著性.Anderson[6]类比热力学提出了漏洞发现模型,但并没有进行有效实验.本文在其基础上进行了相关实验,发现其仅能勉强拟合 8 个软件漏洞发布过程中的 1 个.Gophalakrishna 和 Spafford[7]在其技术报告中研究了漏洞的发现趋势,但没有提出任何的模型.Arbach[8]和 Browne[9]使用了 CERT 的报告,并提出了漏洞总数和时间的平方成正比的观点.Browne[9]提出了漏洞利用模型 VEM.Alhazmi 在前人的基础上总结出了 AML 模型[4,10,11].之后,Alhazmi 对多种漏洞类型进行分类研究,发现对于不同的漏洞类型,漏洞数量与时间关系也基本遵循 AML 模型[12].Kim[13]在 AML 模型的基础上对软件多个版本进行漏洞发布规律的分析,提出了软件多版本模型.AML 模型将软件发布后的生命周期分为 3 个阶段:初期时,漏洞发布过程较为平缓;之后进入快速增长期阶段;最后,发布速度再次进入平缓期.AML 模型相对之前的模型进步许多,具有更广泛的适用性,但它仍然存在其局限性.通过观察,软件漏洞发布总数随时间可能存在多个快速增长期,但 AML 模型并没有进行多快速增长期过程的描述.这些可以通过后文实验加以验证.到目前为止,没有一个模型进行了多快速增长期过程的描述,我们可以把一个快速增长期称为一次突发过程.

从软件工程角度,缺陷预测主要有 3 类模型:对软件开发过程中全生命周期的缺陷分布进行预测的 Rayleigh 模型;主要针对测试阶段,尤其是验收阶段的缺陷分布的指数分布模型和针对发现缺陷和缺陷隔离的 S 曲线分布模型[2].与这几类模型不同,我们的模型是描述软件发布后,漏洞发布数量与发布时间之间的关系;同时,我们的模型引入周期概念,可以描述软件发布后的生命周期中出现多个快速增长期的情况.

漏洞发布的突发过程是有理可循的.漏洞检测手段的每一次技术进步或者某一类典型软件漏洞的发现,都可能引起具有一类共性的漏洞被同时发现,从而使得漏洞发布总数量在短期内快速增长,继而进入一个增长期.这与软件本身以及人们使用软件的普遍程度等多项因素均存在关系.单一增长期模型是多增长期模型的特殊情况:由于软件生命周期的影响,某些软件在还未进入第 2 个增长期时即已脱离市场,被新的版本或新的软件所替代,从而没有呈现出第 2 增长期的特征.这也是 AML 等单一增长期模型能够成功拟合部分软件的原因.因此,融入多个增长期的模型能够更准确地对漏洞发布过程进行描述与预测.本文引入周期概念,提出了一种多周期的漏洞发布模型,描述了漏洞发现数量与时间之间的关系,预测漏洞发布过程;可以对多阶段漏洞突发过程进行描述,扩展了原单一增长过程的漏洞发布预测模型,增大了现有模型的适用范围.实验表明,较现有模型,本模型更符合漏洞发布的规律,应用范围更广.

本文第 1 节详细介绍多周期的漏洞发布预测模型并给出相关参数的计算方法.第 2 节使用多个版本的 Windows 操作系统进行实验分析.最后做出总结.

1 漏洞发布时间预测模型

1.1 模型的建立

为了建立漏洞发布预测模型,需要分析漏洞发布过程的一些特性.

定义 1(漏洞总数). 一款软件客观存在的漏洞个数,用 V 表示,V 也是能被发现漏洞数量的上限.

定义 2(已知漏洞). 一款软件截至时间 t 时被公开的漏洞数量,用 $\Omega(t)$ 表示.因此 $\Omega(t)<V$.

由此可以定义漏洞发布预测模型应具备的特征,包括有界性、递增性、变化率特征和周期性.

特征 1(有界性). $\lim_{t\to\infty}\Omega(t)<\varepsilon$.

考虑到漏洞总数的确定性,所以易得特征 1.

特征 2(递增性). $\forall t_1,t_2(t_2=t_1+1)\to\Omega(t_1)\le\Omega(t_2)$.

由于漏洞发布过程是已知漏洞数量增加的过程,所以漏洞发布函数需满足增函数的特点.

特征 3(变化率趋于 0). $\lim_{t\to\infty}\Omega'(t)=0$.

可由特征 1 和特征 2 得出.此外,考虑到 $\Omega(t)$ 需要描述多个增长期,所以不妨对 $\Omega'(t)$ 引入周期 T.

特征 4(周期性). $\Omega'(t)$ 具有周期因素.

考虑到以上几点,我们分析漏洞发布总数与时间之间的关系,对漏洞发布增长率进行建模,得到表达式:

$$\frac{\mathrm{d}\Omega(t)}{\mathrm{d}t}=\frac{\alpha\sin^2(\beta\cdot t+\phi)}{t^2+\delta},$$

其中,$\Omega(t)$ 表示当时间是 t 时漏洞的总发布数量.上式左边表示漏洞发布数量对时间的增长率,α,β,ϕ,δ 表示各参数.两边同时积分得到:

$$\int_0^u \mathrm{d}\Omega(t)=\Omega(u)-\Omega(0)=\int_0^u\frac{\alpha\sin^2(\beta\cdot t+\phi)}{t^2+\delta}\mathrm{d}t.$$

因为初始时,漏洞发布数量是 0,即 $\Omega(0)=0$,于是得到如下模型:

$$\Omega(u)=\int_0^u\frac{\alpha\sin^2(\beta\cdot t+\phi)}{t^2+\delta}\mathrm{d}t,$$

其中,u 表示某个预定的时刻.图 1 是 CSF 模型的一个示例,虚线表示随时间变化的漏洞发布总数,实线表示随时间变化的漏洞发布速率.

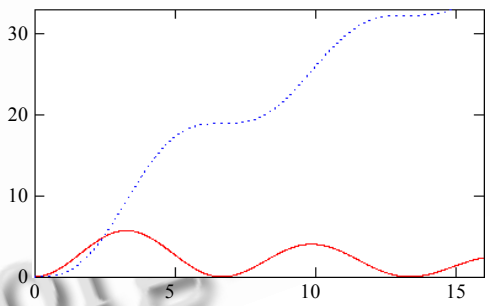

Fig.1 CSF model

图 1 CSF 模型示例图

考虑到漏洞发布过程的离散性,我们使用离散函数进行建模,最终建立如下 CSF 模型:

$$\Omega(t)=\sum_{n=1}^{t}\frac{\alpha\sin^2(\beta\cdot n+\phi)}{n^2+\delta},\ \alpha>0,\delta>0.$$

对 4 个特征的验证:

特征 1(有界性).

证明: $\Omega(t)=\sum_{n=1}^{t}\frac{\alpha\sin^2(\beta\cdot n+\phi)}{n^2+\delta}<\sum_{n=1}^{t}\frac{\alpha}{n^2+\delta}\Rightarrow\lim_{t\to\infty}\Omega(t)<\lim_{t\to\infty}\sum_{n=1}^{t}\frac{\alpha}{n^2+\delta}=\frac{1}{2}\alpha\frac{-1+\delta^{1/2}\pi\cdot\coth(\delta^{1/2}\pi)}{\delta}$,

即取 $\varepsilon=\frac{1}{2}\alpha\frac{-1+\delta^{1/2}\pi\cdot\coth(\delta^{1/2}\pi)}{\delta}$,得证. □

特征 2(递增性).

证明: $\alpha>0,\delta>0\Rightarrow\frac{\alpha\sin^2(\beta\cdot n+\phi)}{n^2+\delta}\ge 0$,

任取 $t_2=t_1+1$,有 $\Omega(t_2)-\Omega(t_1)=\frac{\alpha\sin^2(\beta\cdot(t_1+1)+\phi)}{(t_1+1)^2+\delta}\ge 0$,得证. □

特征 3 和特征 4 显然.

1.2 参数的获取

为了获得最优的参数,首先需要知道模型优劣的评价标准.我们使用国际上常用的卡方检验的方法作为模型好坏的比较标准.卡方系数的计算方法如下:

$$\chi^2=\sum_{i=1}^{n}\frac{(o_i-e_i)^2}{e_i}.$$

这里,o_i 表示观测值,e_i 表示模型的期望值.理论值和观测值越接近,卡方系数就越小,拟合效果也越好.一次模型是否拟合成功,需要验证在特定自由度下,卡方系数是否小于一个特殊的值.我们使用 P 值检验的方法,预设阈值为 5%,即如果计算出的 P 值小于 0.05,则本次拟合过程失败;P 值越接近 1,则效果越好.

这里使用最小二乘法计算各参数的值,使得卡方系数最小.

令 $S(x)=\dfrac{\alpha\sin^2(\beta\cdot x+\phi)}{x^2+\delta}$,目标求 $\min\limits_{S\in\Phi}\left(\sum\limits_i\dfrac{(S(x_i)-Y_i)^2}{S(x_i)}\right)$,其中,$Y_i$ 是观测值,x_i 是对应的发布时间.

再令 $P=\sum\limits_i\dfrac{(S(x_i)-Y_i)^2}{S(x_i)}=\sum\limits_i\left(S(x_i)+\dfrac{Y_i^2}{S(x_i)}-2Y_i\right)$.

为了求 P 的极小值,需要 P 对各参数的偏导数最小.

$$\frac{\partial P}{\partial\alpha}=\sum_i\frac{\partial S}{\partial\alpha}\left(1-\frac{Y_i^2}{S(x_i)^2}\right)=\sum_i\frac{\sin^2(\beta\cdot x_i+\phi)}{x_i^2+\delta}\left(1-\frac{Y_i^2}{S(x_i)^2}\right)=0,$$

$$\frac{\partial P}{\partial\beta}=\sum_i\frac{\partial S}{\partial\beta}\left(1-\frac{Y_i^2}{S(x_i)^2}\right)=\sum_i\frac{\alpha 2\sin(\beta x_i+\phi)\cos(\beta x_i+\phi)x_i}{x_i^2+\delta}\left(1-\frac{Y_i^2}{S(x_i)^2}\right)=0,$$

$$\frac{\partial P}{\partial\phi}=\sum_i\frac{\partial S}{\partial\phi}\left(1-\frac{Y_i^2}{S(x_i)^2}\right)=\sum_i\frac{\alpha\sin(\beta\cdot x_i+\phi)\cos(\beta\cdot x_i+\phi)}{x_i^2+\delta}\left(1-\frac{Y_i^2}{S(x_i)^2}\right)=0,$$

$$\frac{\partial P}{\partial\delta}=\sum_i\frac{\partial S}{\partial\delta}\left(1-\frac{Y_i^2}{S(x_i)^2}\right)=\sum_i\frac{\alpha\cdot\sin^2(\beta\cdot x_i+\phi)}{(x_i^2+\delta)^2}\left(1-\frac{Y_i^2}{S(x_i)^2}\right)=0.$$

将 $S(x)=\dfrac{\alpha\sin^2(\beta\cdot x+\phi)}{x^2+\delta}$ 代入并进行化简,得:

$$\alpha^2\sum_i\frac{\sin^2(\beta\cdot x_i+\phi)}{(x_i^2+\delta)}=\sum_i\frac{Y_i^2(x_i^2+\delta)}{\sin^2(\beta\cdot x_i+\phi)} \tag{1}$$

$$\alpha^2\sum_i x_i\frac{\sin(2(\beta x_i+\phi))}{(x_i^2+\delta)}=\sum_i x_i\frac{Y_i^2(x_i^2+\delta)\sin(2(\beta x_i+\phi))}{\sin^4(\beta\cdot x_i+\phi)} \tag{2}$$

$$\alpha^2\sum_i\frac{\sin(2(\beta x_i+\phi))}{(x_i^2+\delta)}=\sum_i\frac{Y_i^2(x_i^2+\delta)\sin(2(\beta x_i+\phi))}{\sin^4(\beta\cdot x_i+\phi)} \tag{3}$$

$$\alpha^2\sum_i\frac{\sin^2(\beta\cdot x_i+\phi)}{(x_i^2+\delta)^2}=\sum_i\frac{Y_i^2}{\sin^2(\beta\cdot x_i+\phi)} \tag{4}$$

通过观察,上式四元方程组可以转化为三元方程组简化求解.将α提取出,并将公式(1)与公式(4)、公式(2)与公式(3)、公式(1)与公式(3)分别联立合并,即可转化为三元方程组,从而简化计算:

$$\begin{cases}\sum\limits_i\dfrac{\sin^2(\beta\cdot x_i+\phi)}{(x_i^2+\delta)}\sum\limits_i\dfrac{Y_i^2}{\sin^2(\beta\cdot x_i+\phi)}=\sum\limits_i\dfrac{\sin^2(\beta\cdot x_i+\phi)}{(x_i^2+\delta)^2}\sum\limits_i\dfrac{Y_i^2(x_i^2+\delta)}{\sin^2(\beta\cdot x_i+\phi)}\\ \sum\limits_i x_i\dfrac{\sin(2(\beta x_i+\phi))}{(x_i^2+\delta)}\sum\limits_i\dfrac{Y_i^2(x_i^2+\delta)\sin(2(\beta x_i+\phi))}{\sin^4(\beta\cdot x_i+\phi)}=\sum\limits_i\dfrac{\sin(2(\beta x_i+\phi))}{(x_i^2+\delta)}\sum\limits_i x_i\dfrac{Y_i^2(x_i^2+\delta)\sin(2(\beta x_i+\phi))}{\sin^4(\beta\cdot x_i+\phi)}\\ \sum\limits_i\dfrac{\sin^2(\beta\cdot x_i+\phi)}{(x_i^2+\delta)}\sum\limits_i\dfrac{Y_i^2(x_i^2+\delta)\sin(2(\beta x_i+\phi))}{\sin^4(\beta\cdot x_i+\phi)}=\sum\limits_i\dfrac{\sin(2(\beta x_i+\phi))}{(x_i^2+\delta)}\sum\limits_i\dfrac{Y_i^2(x_i^2+\delta)}{\sin^2(\beta\cdot x_i+\phi)}\end{cases}.$$

注意到 x_i 是漏洞发布的时间,可以以序数关系代入,即 $x_i=i$,将其带入上式并再次化简,得

$$\begin{cases}\sum_{i,j}\dfrac{\sin^2(\beta\cdot i+\phi)Y_j^2(i-j)(i+j)}{(i^2+\delta)^2\sin^2(\beta\cdot j+\phi)}=0\\ \sum_{i,j}\dfrac{\sin(2(\beta i+\phi))\sin(2(\beta j+\phi))(j^2+\delta)Y_j^2(i-j)}{(i^2+\delta)\sin^4(\beta\cdot j+\phi)}=0\\ \sum_{i,j}\dfrac{\sin(\beta i+\phi)(j^2+\delta)Y_j^2(\sin\beta(i-j))}{(i^2+\delta)\sin^3(\beta j+\phi)}=0\end{cases}.$$

此方程组可以使用准牛顿法进行近似求解,求解完成后,即可得到α的值为

$$\alpha=\sqrt{\sum_i\frac{Y_i^2(i^2+\delta)}{\sin^2(\beta\cdot i+\phi)}\Big/\sum_i\frac{\sin^2(\beta\cdot i+\phi)}{(i^2+\delta)}}.$$

2 实验与分析

2.1 有效性测试

为了对 CSF 模型的有效性进行测试,我们选用 8 个版本的 Windows 操作系统进行实验.漏洞数据采自 National Vulnerability Database 的漏洞库[14],该漏洞库对各个漏洞进行分类、评级,同时也是国际上常用的漏洞数据库.为了便于对比,这里选用此数据库作为漏洞数据源.

使用 MATLAB 2006b 进行不等式的求解以及卡方系数计算和比较.

表 1 是 CSF 模型拟合效果表.最左边一栏表示进行测试的各个软件,α,β,ϕ,δ表示各参数的取值,*DF* 是自由度,χ^2 表示卡方系数,最后 3 栏分别是 *P* 值、是否具备有效性和增长期数量.

Table 1 Goodness of fit results for Windows datasets on CSF model

表 1 CSF 模型对 Windows 系列数据的拟合效果表

System	α	β	ϕ	δ	*DF*	χ^2	*P*-Value	Fit result	Number of growth phase
Win 95	29 330	0.228	1.83	6 626	24	3.885 2	1	E^*	2
Win 98	14 246	0.308	6.18	2 522	31	12.607 7	0.998 6	*E*	3
Win 2000	26 203	0.044	0.004	1 148	44	29.953 1	0.947 6	*E*	1
Win Me	26 420	0.422	1.977	4 090	17	8.197 9	0.962 1	*E*	2
Win XP	25 488	0.064	0.011	855	25	24.766 9	0.475 5	*E*	1
Win 2003	47 300	0.107	0.009	2 990	22	12.510 5	0.945 9	*E*	1
Win Vista	3 283	0.57	11.28	585	16	9.104 4	0.909 1	*E*	3
Win NT4	23 325	0.05	0.11	2 467	52	57.492 2	0.279 1	*E*	1

E^*: Valid

从表 1 看出,CSF 模型对于各个软件均具有效性.同时观察增长期数,半数软件具有多周期形态分布,充分证明了将多周期概念引入本模型的正确性.值得注意的是,对于其他仅有单次增长期的模型,可以作为 CSF 模型的一个单周期特殊情况对待,可见本模型具有普适性.以下两图分别是 CSF 模型对于多周期分布和单周期分布的拟合效果.图 2 是 Vista 的效果,图 3 是 Windows XP 的效果.

2.2 参数初值的估算

在参数的求解过程中,参数初值的选取对拟合过程有着重要意义.同时,获取相对准确的初值可以在未取得任何漏洞发布数据时对软件的漏洞发布过程进行估算.本节将分析各参数初值与模型结果之间的关系,提出参数初值的估算方法,为后续的拓展研究打下基础.

表 2 分析了 CSF 模型各参数与实际数据间的关系.通过观察,可以总结出各参数的初值选取规则.

根据模型的定义,易知模型的周期为π/β,假设需要估算的时间长度为 *DF*,则 $DF/(\pi/\beta)$的取值与增长期数相

关.从表 2 中的第 2 列和第 3 列也可以观察到这点.通过对增长期数的预估计,可以选择β的初值.β取值越小,函数周期值越大,对于同样长度的时间范围,增长期数也越小.观察表 1 结果,增长期数在范围 1~3 之间,因此在计算初值的时候可以选取增长期数为 2 进行初值计算,并在获得一定数据后进行修正.例如,估算生命周期为 6 年的软件,如果选每 3 个月为一次记录点,因此其自由度为 24.假设其生命周期中有两次漏洞快速增长阶段,即每 3 年漏洞快速增长一次,则可以取$\beta=T*\pi/DF=0.26$ 为初值进行计算,其中,T 为增长期数.

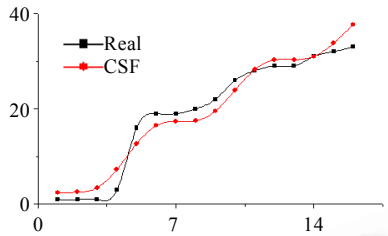

Fig.2 Fitting Windows Vista cumulative vulnerabilities to the CSF model

图 2 Windows Vista CSF 模型拟合效果图

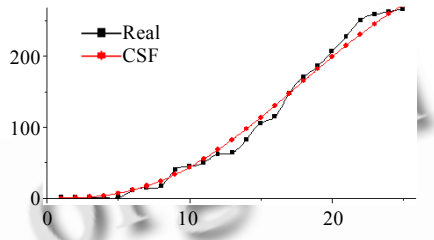

Fig.3 Fitting Windows XP cumulative vulnerabilities to the CSF model

图 3 Windows XP CSF 模型拟合效果图

Table 2 Relationship of CSF model parameters

表 2 CSF 模型各参数关系表

System	$DF/(\pi/\beta)$	Number of growth phase	ϕ	$\frac{D}{F}$	$\alpha/\delta*DF$	Number of vulnerabilities
Win Vista	2.904 459	3	11.28	16	89.791 45	33
Win 95	1.742 675	2	1.83	24	106.236	45
Win Me	2.284 713	2	1.977	17	109.814 2	56
Win 98	3.040 764	3	6.18	31	175.109 4	89
Win 2003	0.749 682	1	0.009	22	348.026 8	185
Win NT4	0.828 025	1	0.11	52	491.649 8	214
Win XP	0.509 554	1	0.011	25	745.263 2	267
Win 2000	0.616 561	1	0.004	44	1 004.29 6	357

不妨将 CSF 模型中的求和看作积分,这样在时间长度为 DF 的范围内,可以将漏洞数量看作是 $f(n)=\frac{\alpha\sin^2(\beta\cdot n+\phi)}{n^2+\delta}$ 在 DF 范围内的面积.当δ=1000 量级,在初始时候 $n^2+\delta\approx\delta$.设 DF 范围内的漏洞数量为 V,因为 $\max(\sin^2(\beta\cdot n+\phi))=1$,可以近似认为$\alpha\cdot DF/\delta=\rho\cdot V$,其中,$\rho$是比例系数.从表 2 中可以看出$\rho$近似为 2,因此计算初值时可取$\alpha/\delta=\rho\cdot V/DF$.观察表 1,对于类似于 Windows 的大型程序,漏洞数量平均取值为 100 上下,对于小型软件可以相对减少.仍然使用上例,对于在自由度为 24 的范围内,假设漏洞数量为 100,可取α/δ=8.33.观察表 1,α通常可取 20 000,因此δ可取 2 400.

ϕ表示偏移量,由于 $\Omega(1)=\frac{\alpha\sin^2(\beta\cdot 1+\phi)}{1^2+\delta}$,可取 $\phi=\arcsin\left(\sqrt{\frac{(1+\delta)\cdot\Omega(1)}{\alpha}}\right)-\beta$,其中,$\Omega(1)$表示第 1 个时间单位内的漏洞发布数量,由此可得$\phi$的初值.对于上例,假设第 1 个时间单位内的漏洞发布数量为 2,α,β,δ使用之前计算的结果,通过计算可得ϕ=0.252.

获取各参数的初值后,可以使用这些值进行前期预测过程的估算.此后,每当有新的漏洞数据公布,即可使用新的数据对模型各参数进行修正计算.

2.3 现有模型比较

本节使用多种流行的漏洞预测模型进行对比,观察各模型对软件漏洞发布过程的拟合程度.这里仍然使用 P 值检验的方法对各模型进行检验,相应阈值设为 0.05.常用的模型包括 AT 模型、AML 模型、RQ 模型、RE 模型、LP 模型和 LM 模型.

AT 模型起初用于软件可靠性[15],之后,Anderson 将它应用于漏洞预测研究[6].AT 模型认为漏洞总数是时间的对数表达式:

$$\Omega(t)=\frac{k}{\gamma}\ln(Ct),$$

其中,C 是积分常量,k 和γ是参数,t 是时间.该模型认为漏洞总是随着时间呈现对数增长.

AML 模型认为漏洞总数与时间的关系如下[1,11]:

$$\Omega(t)=\frac{B}{BC\mathrm{e}^{-ABt}+1},$$

其中,A,B,C 均为参数.AML 模型认为开始时漏洞数量会缓慢增加,之后进入一个快速增长期,漏洞数量随着时间加速增加,最后再次进入平缓期,漏洞的发布过程结束.

Rescola 二次方模型(RQ)[1]认为漏洞的发布速率是线性关系,因此得到漏洞的总数与时间关系如下:

$$\Omega(t)=\frac{Bt^2}{2}+Kt.$$

Rescola 使用了 SRGM 模型进行数据拟合,得到指数关系模型(RE)[1]:

$$\Omega(t)=N(1-\mathrm{e}^{-\lambda t}).$$

对数泊松模型(LP)也被称为 Musa-Okomoto 模型[16],它认为漏洞总数满足如下关系:

$$\Omega(t)=\beta_0\ln(1+\beta_1 t).$$

这里,β_1 和β_2 是回归系数,它认为漏洞总数是对数增长的.

线性模型(LM)认为漏洞总数与时间满足如下线性关系:

$$\Omega(t)=vt+u.$$

表 3 是 CSF 模型与其他各种模型的拟合效果对比表,漏洞数据取自国际上常用的 National Vulnerability Database 的漏洞库[14].

Table 3 Goodness of fit results for Windows dataset on seven models

表 3 7 个模型对 Windows 系列数据的拟合效果表

	AT		LP		LM		RE		RQ		AML		CSF	
	PV[1]	FR[2]	PV	FR	PV	FR	PV	FR	PV	FR	PV	FR	PV	FR
Win 95	0	NE[#]	0.957 9	E^*	0.000 3	NE	0.958 8	E	0.960 0	E	0.999 6	E	1	E
Win 98	0	NE	0.989 5	E	0.996 6	E	0.987 3	E	0.983 6	E	0.558 9	E	0.998 6	E
Win 2000	0	NE	0	NE	0	NE	0	NE	0	NE	0.662 8	E	0.947 6	E
Win XP	0	NE	0	NE	0	NE	0	NE	0.000 1	NE	0.113 4	E	0.475 5	E
Win NT4	0	NE	0	NE	0	NE	0	NE	0	NE	0	NE	0.279 1	E
Win Me	0	NE	0.887 2	E	0.003 8	NE	0.887 1	E	0.993 4	E	0.991 7	E	0.962 1	E
Win 2003	0	NE	0	NE	0.000 4	NE	0	NE	0	NE	0.771 7	E	0.945 9	E
Vista	0.091 8	E	0.293 4	E	0.340 5	E	0.295 9	E	0.299 5	E	0.715 2	E	0.909 1	E

PV[1]: P-Value; FR[2]: Fit result; E^*: Valid; NE[#]: Invalid

从表 3 看出:AT 模型仅能勉强拟合 Vista 系统;LM,LP,RE,RQ 模型仅能拟合不超过一半的系统;相对较好的 AML 模型对于 Win NT4 操作系统仍然不能拟合;CSF 模型能够应对所有的系统,通用性更强.从准确性角度来看,CSF 模型中 75%的 P 值大于 0.9,而 AML 模型仅有 25%.这也说明了 CSF 模型的拟合效果更为理想.

究其原因,CSF 模型具备描述多周期的能力,其他模型仅能描述单一周期的情况,不具备漏洞发布过程的特征 4,RQ,RE,LM 模型甚至不满足于漏洞发布过程的特征 1 和特征 3.因此,对于具备多周期特征的漏洞发布过程,其他两类模型描述能力会下降,甚至不具备描述能力.这也是这类模型在显著性测试中失败的原因.

在 P 值较为接近的情况下,图形的曲线形式仍然会有很大的不同.图 4 与图 5 以 Win 95 为例加以说明.图中,LP,RE,RQ,AML 与 CSF 模型的 P 值均有效,即都具有拟合效果.但 LP,RE,RQ 模型图形几乎呈一直线,并不能将真实漏洞发布过程正确描述;AML 与 CSF 模型较为正确拟合出漏洞发布过程.

从表 1 中得出,Windows 95 具有两个增长期,但是由于第 1 个增长期并不明显,所以 AML 模型才能较为准确的进行拟合.图 6 与图 7 选用具有三增长期的 Windows 98 与 Windows Vista 作 AML 模型与 CSF 模型的对比分析.图 6 是 Windows 98 的 CSF 模型与 AML 模型拟合图,虽然两者拟合结果都是有效的,但是从图中可以明显看出,CSF 模型成功拟合出漏洞发布过程中多个增长期,而 AML 模型仅拟合出一条增长的曲线.图 7 是 Windows Vista 的 CSF 模型与 AML 对比图,图中仍然可以发现,AML 模型把 Vista 的增长过程缩减为一个,而 CSF 模型识别出 Vista 漏洞发布的多个增长过程.

实验结果表明,CSF 模型更具通用性与准确性,在曲线形态方面也更为接近真实的漏洞发布过程.

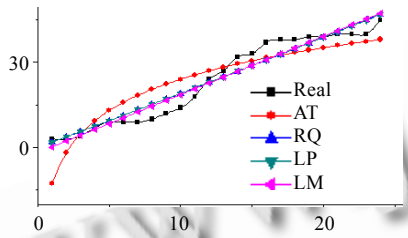

Fig.4 Fitting Windows 95 cumulative vulnerabilities to the AT, RQ, LP, LM model

图 4 Windows 95 AT,RQ,LP,LM 模型拟合效果图

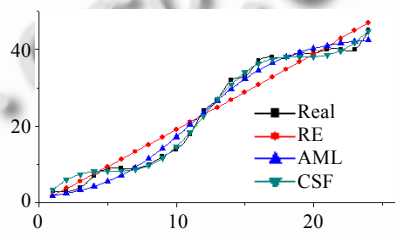

Fig.5 Fitting Windows 95 cumulative vulnerabilities to the RE, AML, CSF model

图 5 Windows 95 RE,AML,CSF 模型拟合效果图

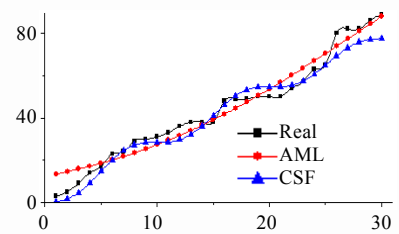

Fig.6 Fitting Windows 98 cumulative vulnerabilities to the CSF, AML model

图 6 Windows 98 CSF,AML 模型拟合效果图

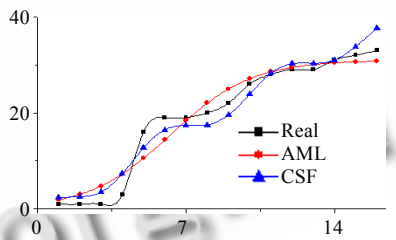

Fig.7 Fitting Windows Vista cumulative vulnerabilities to the CSF, AML model

图 7 Windows Vista CSF,AML 模型拟合效果图

3 结 论

本文针对漏洞发布过程中出现的多周期特点,在漏洞预测模型中引入了发布周期概念,提出了多周期的漏洞发布预测模型.本文同时给出了各参数初值的估算方法,完成了序列对比实验.实验结果表明,本模型成功拟合了 AML 等常用模型不能拟合的软件漏洞发布过程,更符合漏洞的发布规律,增加了漏洞预测的适用范围,提高了预测结果的准确性.

在未来的工作过程中,我们将通过对软件漏洞的形成原因作进一步深入的分析,对模型的精确度进一步修正.同时,在参数初值的选取基础上更深入地研究各参数的意义及其相互之间的关系,提高预测结果的准确度.此外,对漏洞进行分类,研究各种类别下漏洞发布数量与时间的关系也是我们下一步的工作重点之一.

References:

[1] Alhazmi O. Assessing vulnerabilities in software systems: A quantitative approach [Ph.D. Thesis]. Fort Collins: Colorado State University, 2006.

[2] Wang Q, Wu SJ, Li MS. Research on software defect prediction. Journal of Software, 2008,19(7):1565−1580 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/19/1565.htm [doi: 10.3724/SP.J.1001.2008.01565]

[3] Takahashi M, Kamayachi Y. An empirical study of a model for program error prediction. IEEE Trans. on Software Engineering, 1989,15(1):82−86. [doi: 10.1109/32.21729]

[4] Alhazmi OH, Malaiya YK, Ray I. Measuring, analyzing and predicting security vulnerabilities in software systems. Computers & Security, 2007,26(3):219−228. [doi: 10.1016/j.cose.2006.10.002]

[5] Rescorla E. Is finding security holes a good idea? IEEE Security & Privacy, 2005,3(1):14−19. [doi: 10.1109/MSP.2005.17]

[6] Anderson R. Security in open versus closed systems—The dance of Boltzmann, Coase and Moore. In: Proc. of the Conf. on Open Source Software Economics. MIT Press, 2002. 1−15. http://www.cl.cam.ac.uk/~rja14/Papers/toulouse.pdf

[7] Gopalakrishna R, Spafford EH. A trend analysis of vulnerabilities. West Lafayette: Purdue University, 2005. 1−13.

[8] Arbaugh WA, Fithen WL, McHugh J. Windows of vulnerability: A case study analysis. Computer, 2000,33(12):52−59. [doi: 10.1109/2.889093]

[9] Browne HK, Arbaugh WA, McHugh J, Fithen WL. A trend analysis of exploitations. In: Proc. of the IEEE Symp. on Security and Privacy. Oakland: IEEE Press, 2001. 214−229. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=924300&tag=1

[10] Alhazmi OH, Malaiya YK. Modeling the vulnerability discovery process. In: Proc. of the 16th IEEE Int'l Symp. on Software Reliability Engineering. Chicago, 2005. 129−138. http://ieeexplore.ieee.org/search/srchabstract.jsp?tp=&arnumber=1544728&queryText%3DModeling+the+vulnerability+discovery+process%26openedRefinements%3D*%26searchField%3DSearch+All

[11] Alhazmi OH, Malaiya YK, Ray I. Security vulnerabilities in software systems: A quantitative perspective. In: Proc. of the Ann. IFIP WG11.3 Working Conf. on Data and Information Security. Berlin, Heidelberg: Spriger-Verlag, 2005. 281−294. http://www.springerlink.com/content/60xvy1h25jrm3w2g/

[12] Alhazmi OH, Woo SW, Malaiya YK. Security vulnerability categories in major software systems. In: Proc. of the IASTED Int'l Conf. on Communication, Network and Information Security. Cambridge, 2006. 138−143. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.61.6537&rep=rep1&type=pdf

[13] Kim J, Malaiya YK, Ray I. Vulnerability discovery in multi-version software systems. In: Proc. of the 10th IEEE High Assurance Systems Engineering Symp. Dallas: IEEE Computer Society, 2007. 141−148. http://ieeexplore.ieee.org/search/srchabstract.jsp?tp=&arnumber=4404736&queryText%3DVulnerability+discovery+in+multi-version+software+systems%26openedRefinements%3D*%26searchField%3DSearch+All

[14] National vulnerability database. 2008. http://nvd.nist.gov

[15] Brady RM, Anderson RJ, Ball RC. Murphy's law, the fitness of evolving species, and the limits of software reliability. Cambridge: University of Cambridge, Computer Laboratory, 1999. 4−14. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.19.9470&rep=rep1&type=pdf

[16] Musa JD, Okumoto K. A logarithmic poisson execution time model for software reliability measurement. In: Proc. of the 7th Int'l Conf. on Software Engineering. Orlando: IEEE Press, 1984. 230−238. http://portal.acm.org/citation.cfm?id=800054.801975

附中文参考文献:

[2] 王青,伍书剑,李明树.软件缺陷技术研究.软件学报,2008,19(7):1565−1580. http://www.jos.org.cn/1000-9825/19/1565.htm [doi: 10.3724/SP.J.1001.2008.01565]

陈恺(1982−),男,江苏南京人,博士生,主要研究领域为信息安全.

聂楚江(1983−),男,博士生,主要研究领域为信息安全.

冯登国(1965－),男,博士,研究员,博士生导师,CCF 高级会员,主要研究领域为密码学与信息安全.

苏璞睿(1976－),男,博士,副研究员,主要研究领域为恶意代码分析与防范.

张晓菲(1974－),女,博士,主要研究领域为信息安全,可信计算.