基于市场占有率的操作系统安全漏洞检测模型

吴彩华[1] 刘俊涛[2] 朱小冬[1] 叶 飞[1]
(军械工程学院6系维修工程研究所 石家庄 050003)[1]
(军械工程学院5系软件工程教研室 石家庄 050003)[2]

摘 要 操作系统等系统软件中的安全漏洞本质上是一种没有满足软件安全性的缺陷。对安全漏洞的检测过程进行深入研究能够使安全测试人员合理分配测试资源,更准确地评估软件的安全性。深入分析了影响操作系统软件安全漏洞检测的因素,认为安全漏洞检测速度与软件的市场占有率、已发现漏洞数和未发现漏洞数成正比。在此基础上建立了基于市场占有率的漏洞检测模型。该模型表明:在软件发布之前只会暴露少量安全漏洞;某些安全漏洞最终不会被检测到。这两个结论已被实际的数据证实。最后用提出的模型分析了三种流行操作系统的漏洞检测数据集。与同类模型相比,模型具有更好的拟合能力与预测能力。

关键词 安全漏洞,安全性评估,漏洞检测
中图法分类号 TP311.5 **文献标识码** A

Vulnerability Discovery Model Based on Market Share for Operating Systems

WU Cai-hua[1] LIU Jun-tao[2] ZHU Xiao-dong[1] YE Fei[1]
(Maintenance Engineering Institute, Department Six, Mechanism Engineering College, Shijiazhuang 050003 China)[1]
(Sect. Software Engineering, Department Five, Mechanism Engineering College, Shijiazhuang 050003, China)[2]

Abstract Essentially, vulnerability is a kind of software defect dissatisfying the security requirements. Research on the process of vulnerability discovery can help the security testers assign the resource correctly and then evaluate the security of the system accurately. The factors influencing the vulnerability discovery were analyzed, and then it is concluded that the change rate of the cumulative number of vulnerabilities is in direct proportion to the market share of the software, number of discovered vulnerabilities and the number of undiscovered vulnerabilities. It is concluded that the vulnerability discovery model based on market share for operating systems was proposed. Only a few vulnerabilities are discovered while the software is published and some of the vulnerabilities can never be discovered, which is proved in practice. Finally, the vulnerability discovery data of three popular operating systems were analyzed using the proposed model. Compared with the similar model, the proposed model is better at fitting and prediction.

Keywords Vulnerability, Security evaluation, Vulnerability discovery

1 引言

近年来,操作系统等系统软件中的安全隐患越来越引起人们的重视。大量的研究工作关注如何发现软件系统中的安全漏洞[1-3]。然而,对如何定量地分析、评估、预测安全漏洞的检测过程却研究得很少。对此问题进行深入研究可以指导软件安全测试人员合理地分配人力、物力、时间等资源;也可以使开发者定量地评估软件安全性,据此选择软件的最佳发布或升级时机;还可以方便最终用户依据软件的安全性选择软件产品,以减小其受到攻击的风险。

软件的安全漏洞(Vulnerability)指的是计算机的软件系统或组件在设计、编码、配置和使用过程中的错误造成的缺陷,恶意攻击者能够利用这个缺陷对系统资源进行未授权的访问或滥用,违背系统的安全策略,引发安全问题。所以,安全漏洞实际上是一种特殊的软件缺陷,它使得攻击者可以绕过系统的安全措施[4]。换句话说,软件的安全漏洞可以被看作是软件没有满足安全性需求的缺陷,因此软件安全性评估与软件可靠性评估具有一定的相似性。软件可靠性增长模型(SRGM)在评估软件可靠性中已得到广泛的应用。由于软件的安全漏洞本质上是一种软件缺陷,因此可以借用可靠性增长模型的研究方法与成果来分析软件安全漏洞的检测。此类模型大多假设软件故障的发现速度与软件中剩余故障数和故障检测率有关。文献[5]提出软件中剩余故障被检测到的概率是一个随时间变化的连续函数。文献[6]提出了基于响铃形故障检测率函数的软件可靠性增长模型,认为软件的故障检测率是一个先增后减的函数。

到稿日期:2008-05-05 本文受十一五国防预先研究项目(项目名称:软件密集型装备保障技术,项目编号:513270104)资助。
吴彩华(1980—),女,博士研究生,研究方向为软件测试、软件可靠性评估、软件安全性评估,E-mail:wucaihua_1999@yahoo.com.cn;刘俊涛(1979—),男,硕士,讲师,研究方向为软件测试、软件可靠性评估、软件安全性评估;朱小冬(1964—),男,博士,教授,博士生导师,研究方向为软件测试、软件保障、软件安全性评估;叶 飞(1979—),男,博士,讲师,研究方向为软件测试、软件可靠性维护性评估。

另一方面，与软件故障的检测相比，安全漏洞的检测又有其特殊性。通常，软件故障是在软件发布之前的测试阶段检测出来的，而软件安全漏洞往往是在软件发布以后，经过一段时间的使用才被人们发现。而软件的运行环境比软件的测试环境更复杂，因此影响软件安全漏洞检测的因素比影响软件故障检测的因素更多、更复杂。例如，Alhazmi 等人认为软件的安全漏洞检测的速度与已发现的安全漏洞和系统中未被发现的漏洞成正比[7,8]，提出了软件安全漏洞检测的对数模型(AML)。该模型能够较好地描述软件的安全漏洞检测过程，具有一定的预测能力。然而，软件的市场占有率也是影响安全漏洞检测的一个重要因素。攻击市场占有率高的软件能够获得更高的收益，因而更易吸引攻击者的注意力，使得此类软件的安全漏洞更快暴露。文献[9]虽然提到了市场占有率对软件安全漏洞检测的影响，却没有对此建模。为了弥补上述模型的不足，本文在分析软件市场占有率对软件安全漏洞检测的影响的基础上，建立了基于市场占有率的软件安全漏洞检测模型，此模型能够很好地描述软件安全漏洞的发现，并具有很好的预测能力。

2 漏洞检测模型

下面首先深入研究影响软件漏洞检测的因素，然后在此基础上建立基于市场占有率的漏洞检测模型。

2.1 影响漏洞检测的因素

根据文献[5,10]的数据，操作系统的安全漏洞大多数是在软件发布以后发现的。而软件故障的检测则主要是在测试阶段，测试阶段的软件使用环境、使用者与软件运行阶段有明显差异。因此，如果按照可靠性增长模型类推，认为软件安全漏洞的检测速度仅仅与剩余漏洞数和漏洞的检测率有关是不够的。文献[8]假设软件漏洞的检测速度和两个因素有关：一个是软件中未被发现的漏洞数量；另一个是软件已经发现的漏洞数量。第一个假设与软件可靠性增长模型中的假设类似。对于第二个假设，作者认为已发现的软件漏洞数量表示市场对软件的关注程度。然而，事实上，已发现的软件漏洞并不能完全表示人们对该软件的关注程度，例如 Windows 2000 系统，截止到 2007 年 12 月底，已发现安全漏洞 345 个，而 Windows XP 则发现安全漏洞 256 个。很明显，我们不能据此认为 Windows 2000 获得了更高的关注。本文认为，软件的被关注程度是由软件的市场占有率来表示的。而已发现的漏洞数一定程度上反映了软件系统的脆弱性，人们往往认为已发现漏洞数较多的软件更容易被成功攻击。

软件的市场占有率是影响软件安全漏洞发现速度的一个重要因素。从漏洞的发现人员来看，发现漏洞的不仅仅是测试人员，还包括大量的最终用户以及攻击者。使用软件的人越多，安全漏洞暴露的机会就越多，由此可以认为软件漏洞检测的速度和软件的用户规模有关。从攻击的收益来看，如果软件的市场规模比较大，攻击者一旦攻击成功，将获得较大的收益。可以说，软件的用户规模一定程度上鼓励了攻击行为。以 Windows 2003 为例，图 1 显示了 2004 年 9 月到 2007 年 9 月 Windows2003 安全漏洞的月平均发现数量，图 2 显示了同一时期 Windows 2003 的市场占有率变化趋势。从图中可以看到，Windows2003 的市场占有率在前 20 个月的时间呈上升趋势，并且在这一段时间里安全漏洞的发现速度也在增加；从第 25 个月开始，Windows 2003 的市场占有率发生了波动，紧接着漏洞的发现速度也发生了波动。对比这两个图，可以看出市场占有率和漏洞的发现速率之间存在某种联系。

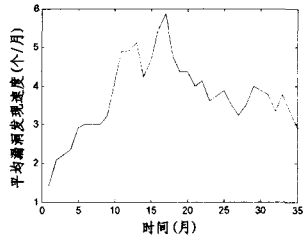

图 1 Windows2003 安全漏洞平均发现速度

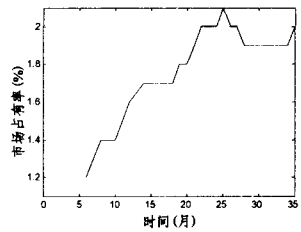

图 2 Windows2003 的市场占有率

2.2 漏洞检测模型

基于以上分析，我们可以假设软件安全漏洞的检测速度与下面 3 个因素成正比：软件的市场占有率、已发现的漏洞数量、软件中剩余的漏洞数量，由此得到下面的微分方程：

$$\frac{dy}{dt}=a(t)\cdot y\cdot(B-y) \tag{1}$$

其中，y 表示累积发现的软件漏洞数，t 是日历时间，$a(t)$ 表示在 t 时刻软件的市场占有率，$B>0$ 是一个常数，表示软件初始漏洞数。

令 $A(t)=\int_0^t a(\tau)d\tau$，称为 t 时刻的累积市场占有率。解方程(1)得：

$$y=\frac{B}{C\cdot e^{-B\cdot A(t)}+1} \tag{2}$$

其中，C 是求解过程中引进的常数。我们称式(2)为基于市场占有率的漏洞检测模型。如果 $a(t)$ 为常数函数，则式(2)退化为 AML 模型。

通过观察 Windows 2000 等流行操作系统的市场占有率变化趋势，我们发现操作系统市场占有率的变化一般经历两个阶段：在软件发布的初期，开发商的推广使得软件的市场规模逐渐扩大；经历一段时间后，一旦新的软件产品推出，老的软件产品的市场占有率会逐步下降。因此，我们用式(3)所示的函数表示一个操作系统市场占有率的变化趋势：

$$a(t)=\left(\frac{b}{1+bt}\right)^2 \tag{3}$$

其中，$b>0$ 是待定的参数。那么累积市场占有率为

$$A(t)=\frac{b^2t}{1+bt} \tag{4}$$

代入式(2)，得到

$$y=\frac{B}{C\cdot e^{-B\cdot\frac{b^2t}{1+bt}}+1} \tag{5}$$

式(5)即为基于市场占有率的漏洞预测模型。利用式(5)分析软件的安全漏洞检测过程需要利用历史数据对该模型中的参数 b,B,C 进行估计。下面我们来分析式(5)的特性。

在软件发布的初期，即当时间 t 趋于 0 时，

$$\lim_{t\to 0}y=\lim_{t\to 0}\frac{B}{C\cdot e^{-B\cdot\frac{b^2t}{1+bt}}+1}=\frac{B}{C+1}\neq 0 \tag{6}$$

依据式(6)我们得到一个结论：在软件发布之前已经有漏洞暴露出来，这些漏洞是在软件测试过程中发现的。例如 Windows 2000 是在 2000 年 2 月 17 日发布的，在这之前已经暴露出了 30 个安全漏洞[10]。同样，Windows XP 在发布之前

也已经暴露了一些安全漏洞[10]。通常，软件发布初期暴露的漏洞数量不会太多，因此参数 B，C 的值相差不大。

当时间 t 趋于无穷时：

$$\lim_{t\to\infty} y=\lim_{t\to\infty}\frac{B}{C\cdot e^{-B\cdot\frac{b^2 t}{1+bt}}+1}=\frac{B}{C\cdot e^{-B\cdot b}+1}<B \tag{7}$$

依据式(7)我们得到了另一个重要结论：不是所有的漏洞都能暴露出来。原因在于软件漏洞的检测与软件故障的检测的目的不同，软件故障检测的目的是为了在软件发布之前尽可能把软件中的故障全部找到。而攻击者或者最终用户并不是以发现所有软件漏洞为目的。随着市场占有率的降低，一方面，软件的使用者越来越少，漏洞暴露的机会越来越少；另一方面，攻击者不能从攻击行为中获得与以前一样的收益，逐渐失去了攻击兴趣。因此漏洞的发现速率越来越低。最终，当软件产品退出市场后，一些漏洞被遗留在系统中，不为人知。我们用 B' 表示软件的最终发现漏洞数，有

$$B'=\lim_{t\to\infty} y=\frac{B}{C\cdot e^{-B\cdot b}+1} \tag{8}$$

3 结果与讨论

本文用新提出的模型—基于市场占有率的漏洞检测模型分析 Windows 2000，Windows XP 和 Windows 2003 的安全漏洞检测数据集，据此检验模型的拟合能力和预测能力。这些系统的安全漏洞检测数据集可以从 National Vulnerabilities Database[10]得到。

3.1 模型的拟合能力

图 3、图 4 和图 5 分别显示了基于市场占有率的漏洞检测模型对 Windows 2000，Windows XP 和 Windows 2003 安全漏洞检测数据集的拟合结果。从图中可以看到，本文模型对 3 种操作系统的安全漏洞检测数据集的拟合取得了令人满意的效果。按照拟合结果，本文认为这 3 种操作系统安全漏洞的发现过程处于上升阶段，在不久的将来仍然会有一些漏洞暴露出来，使用这些系统的安全风险没有减少。表 1 列出了模型参数的计算结果。

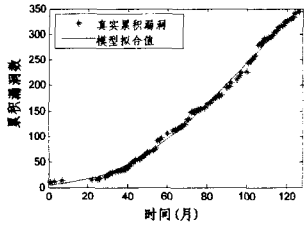

图 3 Windows 2000 的累积安全漏洞

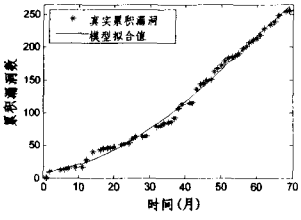

图 4 Windows XP 的累积安全漏洞

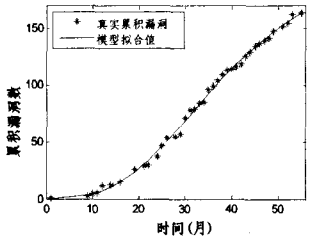

图 5 Windows 2003 的累积安全漏洞

表 2 用均方差、残差平方和、相关系数和卡方系数等标准比较了 AML 与本文模型的拟合效果。这些标准的数值越小，说明拟合的效果越好。从表 2 中可以看到，本文模型对 Windows 2000 和 Windows 2003 的安全漏洞检测数据集的拟合效果明显优于 AML。两个模型对 Windows XP 安全漏洞检测数据集的拟合效果相差不大。

表 1 模型参数计算结果

操作系统	b	B	C
Windows 2000	9.626E-3	943.5	246.9
Windows XP	1.240E-2	682.9	79.77
Windows 2003	4.421E-2	257.1	1779

表 2 本文模型与 AML 拟合结果的比较

系统	模型	均方差 (RMSE)	残差平方和 (SSE)	相关系数 (R)	卡方系数 (χ^2)
Windows 2000	AML	7.442	4818	0.9975	32.59
	本文模型	6.046	3180	0.9983	21.74
Windows XP	AML	5.399	1632	0.9975	25.29
	本文模型	5.758	1856	0.9971	19.81
Windows 2003	AML	3.518	482.7	0.9978	9.063
	本文模型	2.723	289.2	0.9986	3.953

3.2 模型的预测能力

为了验证本文提出模型的预测能力，我们用数据集中前 i 个月的真实数据($\lfloor n/2\rfloor\leqslant i\leqslant n$，$n$ 为数据集中的总月数)，预测数据集中最后一个月的累积漏洞总数，并把预测的结果和真实的结果进行比较。本文采用式(9)所示的平均相对误差 AE 来表示模型的预测能力：

$$AE=\frac{1}{n-\lfloor n/2\rfloor}\sum_{i=\lfloor n/2\rfloor}^{n}\left|\frac{o-e_i}{o}\right| \tag{9}$$

其中，o 是最后一个月累积漏洞总数的真实值，e_i 表示用前 i 个月的真实数据对最后一个月的累积漏洞总数的预测值。AE 的值越小，说明模型的预测能力越好。

图 6 显示了 AML 和本文模型对 Windows 2000 漏洞数据集的预测结果比较，其中横坐标是归一化后的时间，纵坐标是预测结果的相对误差。图 7、图 8 分别显示了对 Windows XP 和 Windows 2003 漏洞数据集的预测结果。从图 6 和图 8 可以看到，本文模型的预测结果在大多数时候都要优于 AML 的预测结果。而从图 7 来看，本文模型的早期预测结果比 AML 的要好，而后期的预测结果则略逊于 AML。

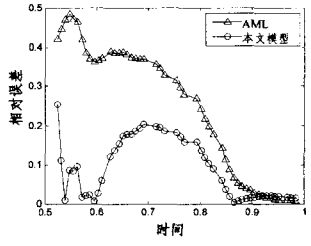

图 6 Windows 2000 漏洞预测的相对误差

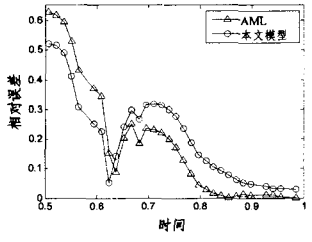

图 7 Windows XP 漏洞预测的相对误差

表 3 显示了 AML 与本文模型对 3 种操作系统安全漏洞预测结果的平均相对误差。可以看到，除对 Windows XP 的漏洞预测外，本文模型的预测结果都要优于 AML。

表 3 漏洞预测的平均相对误差

模型 \ 操作系统	Windows 2000	Windows XP	Windows 2003
AML	24.37%	18.81%	15.88%
本文模型	8.69%	21.01%	11.76%

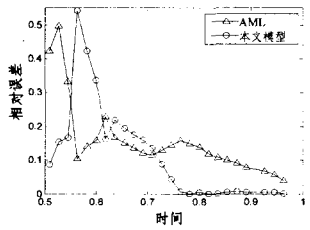

图 8 Windows 2003 漏洞预测的相对误差

结束语 软件的安全漏洞本质上是一种缺陷，因此软件漏洞检测过程与软件故障检测过程有一定程度的相似之处。另一方面，由于软件故障检测和软件漏洞检测的目的、参与人员和发生的时间不尽相同，描述这两个过程的模型又会有一些不同。本文深入分析了影响软件安全漏洞检测的因素。与现有的安全漏洞检测模型不同的是，本文认为软件的市场占有率也是影响软件安全漏洞检测速度的一个重要因素，并在此基础上建立了基于市场占有率的漏洞检测模型。通过分析模型的特性，我们得到了两个重要结论：一是在软件发布之前已经暴露了一些安全漏洞；二是并不是所有的安全漏洞都会被检测出来。这两个结论在实际的数据中得到了验证。我们用本文提出的模型分析了 3 种流行操作系统的漏洞检测数据集，得出了这 3 种操作系统的漏洞检测还处于快速上升阶段的结论。与 AML 相比，大多数情况下本文提出的模型有更好的拟合能力与预测能力。

参 考 文 献

[1] Zhang W F, Tao Xin Li, Jose F. An Analysis of Microarchitecture Vulnerability to Soft Errors on Simultaneous Multithreaded Architectures// IEEE International Symposium on Performance Analysis of Systems & Software, 2007(ISPASS 2007). 2007

[2] Salas P A P, Krishnan P, Ross K J. Model-based Security Vulnerability Testing// 18th Australian Software Engineering Conference 2007(ASWEC 2007). 2007

[3] Savola R K, Kaarina. Practical Security Testing of Telecommunications Software—A Case Study// The Third Advanced International Conference on Telecommunications 2007(AICT 2007). 2007

[4] Schultz J E, Brown D, Longstaff T. Responding to computer security incidents. Lawrence Livermore National Laboratory, 1990. Ftp:\\ftp. cert. dfn. de/pub/docs/csir/ihg. ps. gz

[5] Goel A L, Okumoto K. Time - dependent error - detection rate model for software and other performance measures. IEEE Transaction on Reliability, 1979, 28: 206-211

[6] 刘宏伟，杨孝宗，曲峰，等. 一个基于响铃形故障检测率函数的软件可靠性增长模型[J]. 计算机学报，2005，28(5)：908-913

[7] Alhazmi O H, Malaiya Y K. Application of Vulnerability Discovery Models to Major Operating Systems. IEEE Trans. Reliability, 2008: 14-22

[8] Alhazmi O H, Malaiya Y K, Ray I. Measuring, Analyzing and Predicting Security Vulnerabilities in Software Systems. Computers and Security Journal, 2007: 26(3): 219-228

[9] Alhazmi O H, Malaiya Y K. Measuring and Enhancing Prediction Capabilities of Vulnerabilities Discovery Models for Apache and IIS HTTP Servers// Symp. Software Reliability Eng. 2006: 343-352

[10] National Vulnerability Database. http://nvd. nist. gov/

(上接第 158 页)

员/技能(T)的管理，另外还为收集其它数据构建了接口，以满足其它软件过程度量需求。数据使用层为各级角色定制了度量视图，提供与角色度量需求相适应的数据分析结果。另外，为支持度量数据有效使用，数据使用层还提供了各组织单元的 PCB 和度量活动交流平台。图 4 为企业集成软件过程度量系统启动画面。

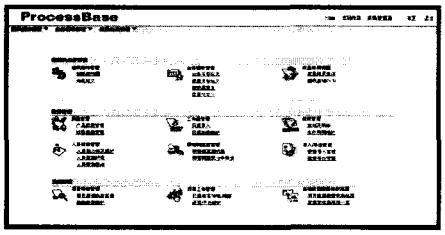

图 4 企业集成软件过程度量系统启动画面

结束语 针对软件企业过程度量活动的特点，作者提出了一个企业集成软件过程度量模型。该模型满足企业度量活动各阶段要求，使企业的度量生产及度量消费协调一致，同时该模型可以支持过程改善活动的顺利开展。

基于企业集成软件过程度量模型，我们开发了一个集成化软件过程度量系统，该系统支持企业过程度量数据的收集、分析及使用，提供项目软件过程动态监控及业务单元经营状态分析，满足企业不同层级角色的过程度量需求。在系统支持下，企业度量活动的目的性及自动化程度均有很大提升。

下一步工作重点是研讨企业软件过程度量信息与企业其它业务信息的融合方法，拓展其应用范围，加强企业间各类信息的互通，减少数据的冗余及不一致性，简化各类角色获取信息的通路，提升软件企业信息化整体水平。

参 考 文 献

[1] CMMI Product Team. Capability Maturity Model Integration (CMMI) for development Version 1. 2. SEI. CMU/SEI-2006-008(ESC-TR-2006-008). 2006. 8

[2] McGarry J, Card D, Jones C, et al. Practical Software Measurement: Objective Information for Decision Makers[M]. Addison-wesley, 2002

[3] Solingen R, Berghout E. The Goal/Question/Metric Method: a Practical Guide for Quality Improvement of Software Development[M]. The McGRAW-HILL Companies, 1999

[4] Basili V R, Caldiera G, Rombach H D. Goal Question Metric Paradigm// Marciniak J J, ed. Encyclopedia of Software Engineering. New York: John Wiley & Sons. Inc., 1994: 528-532

[5] Park R E, Goethert W B, Florac W A. Goal-driven Software Measurement-A Guidebook. SEI CMU/SEI-96-HB-002. 1996. 8

[6] Lawler J, Kitchenham B. Measurement modeling technology. IEEE Software, 2003, 20(3): 42-48

[7] Junlin Huang, Far B H. Intelligent Software Measurement System for Automating the Goal-Question-Metrics Process// Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence(ICTAI'06). 2006

[8] 钱红兵，朱丽娟，曹惠民. 基于 CMM 的软件过程度量系统的研究与设计[J]. 计算机应用研究，2004，21(6)：49-52

[9] 任发科，周伯生，吴超英. 软件度量过程的研究与实施[J]. 北京航空航天大学学报，2003，29(10)：931-934

[10] 王青，李明树，刘霞. 一种支持软件过程控制和改进的主动度量模型[J]. 软件学报，2005，16(3)：407-418