



基于损失量的 G-O 漏洞预测模型及其改进

彭轼, 郭昊, 王涛

(全球能源互联网研究院, 北京 102209)

摘要: 分析了度量漏洞的各个指标, 提出了强安全性的数学定义, 并使用损失量度量和预测漏洞, 解决了软件可信性统一量纲问题。同时, 讨论了损失量出现规律和漏洞数量发现规律之间的相似性, 确定使用预测软件缺陷的模型来预测损失量。通过借鉴经典的 G-O 模型, 建立了软件损失量的预测模型, 即提出了基于损失量的 G-O 漏洞预测模型, 并通过使用实际漏洞库中的数据检验了模型的准确性和实用性。

关键词: 安全漏洞; 损失量度量; G-O 模型; 预测模型

中图分类号: TP311.5

文献标识码: A

doi: 10.11959/j.issn.1000-0801.2015393

G-O vulnerability forecasting model and its improvement based on loss

PENG Shi, GUO Hao, WANG Tao

Global Energy Interconnection Research Institute, Beijing 102209, China

Abstract: Each index to measure the vulnerability was analyzed. A mathematical definition of strong security was proposed, and the loss measurement and forecasting of vulnerability were used to solve the problem of software dependability uniform dimension. At the same time, loss occurrence law and the number of vulnerabilities found similarities between the law were discussed, to determine whether the use of software defect prediction model to predict the amount of loss. By referring to the classical G-O model, the predictive model of software loss was established, namely model was developed to predict the loss of G-O based vulnerabilities, and the accuracy of the model and the practicability of the test by using the actual data in the vulnerability database.

Key words: security vulnerability, loss measurement, G-O model, prediction model

1 引言

软件漏洞是软件缺陷的一种, 其定义众说纷纭。Schultz^[1]定义软件漏洞为“一种缺陷, 使得入侵者可以绕过安全机制”, Pfleeger^[2]将漏洞定义为“安全系统中可被用来引起损失或危害的弱点”, Shin Y^[3]认为“软件漏洞是软件规格、开发或配置中缺陷的一个实例, 它的运行违反潜在或外在的安全策略”。软件缺陷是影响软件安全性的一个

重要因素, 软件安全性是软件可信性的一个重要属性, 国家标准 GB/T 11457 中对软件安全性的定义为: 通过对软件系统进行防护, 阻止针对软件产品的恶意攻击, 使软件系统免于受到恶意的或无意但有害的破坏。

软件漏洞对软件系统有着巨大的危害, 例如, 2011 年 12 月, 中国最大的开发者技术社区——中国软件开发者联盟(CSDN)的系统遭到黑客攻击, 多达 600 万用户的登录名、口令、邮箱等敏感个人信息泄露。因此, 包括政府、软



件供应商、软件使用者以及软件评估机构等相关组织采取建立漏洞库、跟踪漏洞信息等方法,以求降低漏洞的危害。但是,这些都是事后措施,人们尝试使用多种数学模型来预测漏洞,比较有代表性的有热动力模型^[4]、对数泊松模型^[5]、二次模型^[6]和指数模型^[7]等,这些模型以时间为重要参数,试图预测软件漏洞的数量等属性,效果各有千秋。

随着软件理论领域的发展,人们提出了软件可信性这个概念,软件可信性有许多定义^[8-12],它们的共同点在于软件可信性是一个复合的概念,包含了软件几个方面的属性,而这些属性之间如何统一起来反映软件可信性是一个大问题^[13]。以漏洞和故障为例,漏洞有以上提到的几个属性,而故障的属性 and 漏洞不是一一对应的,因此二者不能直接合并。此外,软件可信性包含强可靠性和强安全性^[12],它们的度量量纲也不一样,需要一个统一的度量量纲。

2 损失量度量和预测漏洞

现有实际漏洞库给出的最基本的数据都是发现漏洞的个数、类型和严重程度等数据。在使用这些数据度量软件的安全性时,遇到了两个问题。

第一个问题是现有的漏洞分类方法难以准确描述漏洞对安全性的影响程度。例如,两个网上银行软件可能各有一个漏洞,使得用户的权限泄露,但是其中一个网上银行软件不会造成账户被黑客操作,只是泄露信息,用户必须重置账户信息,银行在此过程中只要支付一定的办公开支即可;而另一个软件系统则造成账户中的钱被取走或消费,银行可能需要赔付客户损失。这两个漏洞即使被认定为属于不同的种类,但是对严重程度的反映并不直观,而且在有些分类方法中,由于都属于权限的泄露,这两个漏洞可能被分为同一级。

第二个问题是为了度量不同软件的可信属性,有必要统一度量的量纲,而现有的漏洞分类方法,要么做不到这一点,要么勉强可以做到,但没有充分地反映软件的可信属性。为了使漏洞的度量能直观、有效地反映软件的可信属性,有必要重新选择度量的量纲。

那么,有没有一个影响因子,既能比较客观地反映出用户对软件系统安全性的信任,又能作为一个统一的量纲来度量软件安全性乃至可信性呢?漏洞所造成的经济损失量符合这个要求。首先,用损失量来描述问题的严重程度,有一目了然的优点。其次,损失量(即钱数)本身就是一个量化的值,而且是统一的。例如,两个不同的漏洞各造成了 10 000 美元的损失,尽管这两个漏洞可能有很大区别,但是 10 000 美元

和 10 000 美元总是相等的。第三,使用损失量度量软件安全性乃至可信性可以让用户抛开细节,直接面对他们关心的方面,从而将软件可信问题的主观因素统一在度量中。

在已知的范围内,学术界和产业界还没有研究过使用损失量度量安全性。因此,为了使用损失量作为度量和预测漏洞发现的量纲,有必要审视一下损失出现的规律,以便选择合适的模型。

首先考察漏洞的出现趋势。通常来讲,一个软件系统发布并投入使用后,漏洞从零开始增长,由于具体情况不同,可能在不同时期有不同的增长速度;随着已发现漏洞被修补,新漏洞的发现速度最终会下降;由于漏洞本质上是软件开发部署过程中系统内部固有的缺陷,所以其数量是有限的,不会无限增长,但也很可能不会全部被发现,而且软件最终会因为更新而下降,所以在一个时间点以后,漏洞数趋近于一个固定值。大概的趋势如图 1 所示。

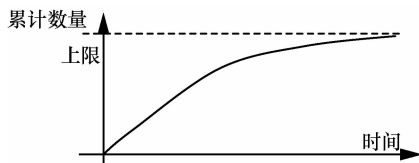


图 1 漏洞数量随时间的变化趋势

漏洞的发现是一个离散的过程,而损失是一个连续的量。损失是由漏洞造成的,但事实上不能得到很精确的记录,而且也不要求预测模型有很精确的结果,只要精确到某个数量级就可以了。所以可以将损失量离散化,比如以 1 万元或 10 万元为单位。虽然损失随着离散的漏洞而出现,但是当考虑损失总量的时候,通过离散化方法,也可以得到类似于图 1 的趋势,如图 2 所示。

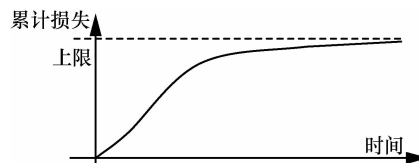


图 2 漏洞造成的损失量随时间的变化趋势

因此,损失量也可以使用那些用于预测软件缺陷的模型来进行预测。由于这些模型中很多是基于概率的模型,所以还要确定漏洞的发现是不是概率问题。

很显然,从不同的角度出发,漏洞的发现规律也不同。由于需要建立的是通用的预测模型,而不是专用的模型,必须选择一个有通用性的,在软件漏洞生命周期内人们所扮演的所有角色中,三方软件质量评测机构最符合要求。

首先,第三方软件质量评测机构通常建立并维护漏洞的数据库,因此,不缺乏模型需要的基础漏洞数据。其次,第三方软件质量评测机构不参与软件的开发,也不会把自己的工作量作为唯一参数,因为他们明白有许多人同样在挖掘漏洞。第三,尽管第三方软件质量评测机构收集的信息不能精确到函数数量或者工作时间这样的精度,也会有一些时延,但是漏洞预测本身也不可能做到非常精确,一定范围内的误差是不可避免的,所以尽管原始数据越翔实越好,但是如果实在难以获得,也没有必要强求。

从第三方软件质量评测机构的角度来看,漏洞的出现有着随机性,而且可以用一个数字来表示这种随机性的大小,因此漏洞的出现可以看作概率问题。

国内外学者提出了许多模型,比较有代表性的有热动力(AT)模型、对数泊松模型、二次模型、指数模型、逻辑模型、线性模型、多周期模型等。在这些模型中,MUSA^[14,15]等人提出的模型考虑了累计的故障数、故障出现率的衰减等因素,因此有十分重要的参考意义。本文将从损失和时间的联系出发,推导出预测模型,并根据实际问题进行改进。

3 漏洞损失量模型

3.1 基本定义及模型

设 $\{M(\tau), \tau \geq 0\}$ 为一个计数过程,表示到时间 τ 为止由漏洞造成的累计损失量。显然:

$$M(0)=0 \quad (1)$$

用 $m(\tau)$ 表示到时间 τ 为止由漏洞造成的累计损失量的期望,为 τ 的有界不递减函数, $m(\tau)=E[M(\tau)]$ 满足以下关系:

$$m(\tau)=\begin{cases} 0, & \tau=0 \\ \alpha, & \tau \rightarrow \infty \end{cases} \quad (2)$$

其中, α 是最终软件中所有漏洞造成损失量的和的期望。

假设在时间 $(\tau, \tau+\Delta\tau)$ 内漏洞造成损失的期望正比于所有未发现漏洞造成损失的和的期望,则:

$$m(\tau+\Delta\tau)-m(\tau)=\beta[\alpha-m(\tau)]\Delta\tau+o(\Delta\tau) \quad (3)$$

其中, β 是常数,且当 $\Delta\tau \rightarrow 0$ 时, $o(\Delta\tau)/\Delta\tau \rightarrow 0$ 。

在式(3)中令 $\Delta\tau \rightarrow 0$,则:

$$m'(\tau)=\alpha\beta-\beta m(\tau) \quad (4)$$

解此微分方程,可得:

$$m(\tau)=\alpha(1-e^{-\beta\tau}) \quad (5)$$

$m(\tau)$ 即 $M(\tau)$ 的均值函数,其密度函数为:

$$\lambda(\tau)=m'(\tau)=\alpha\beta e^{-\beta\tau} \quad (6)$$

$\{M(\tau), \tau \geq 0\}$ 是均值函数为 $m(\tau)$ 、密度函数为 $\lambda(\tau)$ 的泊

松过程, y 是自然数,并且满足:

$$Pr\{M(\tau)=y\}=\frac{[m(\tau)]^y}{y!}e^{-m(\tau)} \quad (7)$$

为了区别于传统的安全度,定义基于损失量的安全度为强安全度,其具体定义为在上一个损失时间一定的条件下,下一段时间内不发生漏洞引起的损失的条件概率。

$T'_i(i=1,2,\dots)$ 表示第 $i-1$ 个损失 m_{i-1} 到第 i 个损失 m_i 的时间间隔的随机变量, $T_i(i=1,2,\dots)$ 表示到第 i 个损失 m_i 的时间的随机变量,则:

$$T_i=\sum_{j=1}^i T'_j=T_{i-1}+T'_i \quad (8)$$

其中, $T_0=0$ 。

给定时间序列 $\tau_1, \tau_2, \dots, \tau_n$,则 $T_1, T_2, \dots, T_n$ 的联合密度函数为:

$$f_{T_1, T_2, \dots, T_n}(\tau_1, \tau_2, \dots, \tau_n)=e^{-m(\tau_n)} \prod_{k=1}^n \alpha\beta e^{-\beta\tau_k} \quad (9)$$

对数似然函数为:

$$L(\alpha, \beta|\tau)=n\ln\alpha+n\ln\beta-\alpha(1-e^{-\beta\tau_n})+\beta\sum_{k=1}^n \tau_k \quad (10)$$

分别对 α, β 求导,并对导数赋0,建立方程组:

$$\begin{cases} \frac{n}{\alpha}=1-e^{-\beta\tau_n} \\ \frac{n}{\beta}=\alpha\tau_n e^{-\beta\tau_n}+\sum_{k=1}^n \tau_k \end{cases} \quad (11)$$

由此用数值方法可求解,解出 α, β 。

3.2 实际数据检验

从中国国家信息安全漏洞库中选取了 Windows XP 操作系统, Adobe Flash Player 和 Mozilla Firefox 从发布起,累计 200 个漏洞的数据。这些漏洞按严重程度被分为 4 类,其中等级 4 最危险。假定严重等级越高损失越大,一个等级对应一个损失量级,在本文的例子中规定严重程度为 1 的漏洞造成 1 万元损失,严重程度为 2 的漏洞造成 2 万元损失,以此类推。将漏洞按发现日期排列,计算出发现日期距软件发布日期相差的天数。同时,假设一天只发现一次损失,因此将同一天内的损失量合并取平均值。

使用前 50 个数据来估算剩下的数据并与实际数据比较,误差在 ± 0.5 之间的结果记为预测成功,计算结果见表 1 ($\alpha=257.0, \beta=0.0036$, 准确率约为 20%)、表 2 ($\alpha=288.0, \beta=0.0021$, 准确率约为 35%)、表 3 ($\alpha=211.0, \beta=0.0061$, 准确率约为 30%)。



表 1 Windows XP 预测结果

序号	损失/万元	天数/日	累计损失/万元	估计值/万元	是否准确
1	3	1 327	128	127.441 6	否
2	3	1 328	131	128.451 5	否
3	2	1 363	133	133.100 7	是
4	3	1 371	136	134.141 5	否
5	1	1 385	137	138.696 4	否
6	1	1 407	138	139.270 1	否
7	2	1 442	140	141.359 8	否
8	2	1 448	142	141.674 0	是
9	2	1 449	144	142.881 9	否
10	4	1 457	148	143.903 8	否

表 2 Flash Player 预测结果

序号	损失/万元	天数/日	累计损失/万元	估计值/万元	是否准确
1	3	2 347	143	142.061 4	否
2	4	2 372	147	146.123 5	否
3	3	2 382	150	148.652 6	否
4	3	2 448	153	154.285 6	否
5	4	2 454	157	156.803 4	是
6	2	2 467	159	159.331 1	是
7	3	2 488	162	161.869 1	是
8	4	2 505	166	164.399 7	否

表 3 Firefox 预测结果

序号	损失/万元	天数/日	累计损失/万元	估计值/万元	是否准确
1	3	728	103	103.760 9	否
2	2	729	105	104.297 7	否
3	2	745	107	106.927 7	是
4	2	766	109	109.059 0	是
5	2	770	111	111.600 9	否
6	2	820	113	113.367 6	是
7	2	826	115	114.381 6	否
8	3	828	118	115.816 1	否
9	2	832	120	118.633 1	否
10	2	835	122	119.640 7	否

4 改进模型

从以上分析可以看到,预测结果并不理想,下面将对预测结果进行分析,改进预测模型,以得到更好的效果。

设 τ_i 为发生第 i 次损失的时间, τ'_i 为此次损失与上一次损失之间的时间间隔,即 $\tau'_i = \tau_i - \tau_{i-1}$,则在时间 $(\tau_{i-1}, \tau_i]$ 中产生损失与 τ'_i 相关, ω 为相关系数。

设 $t = f(\tau) = \tau + \omega\tau'$, 为了表示方便,没有给 t 、 τ 和 τ' 加下标,在实际数据中它们是一一对应的。

为了保证损失量不变,设随机过程 $N(t)$, 规定: $N(t) = N[f(\tau)] = M(\tau)$, $Pr\{N(t) = m\} = Pr\{M(\tau) = m\}$ 。

设 $N(t)$ 的均值函数为 $\mu(t)$, 密度函数为 $\lambda(t)$, 则:

$$\begin{aligned}\mu(t) &= \int \lambda(t) dt \\ &= E[N(t)] = E\{N[f(\tau)]\} \\ &= E[M(\tau)] = \mu(\tau)\end{aligned}\quad (12)$$

这相当于从坐标系 $\{\tau, \mu(\tau)\}$ 转换到坐标系 $\{t, \mu(t)\}$, 由于 t 是 τ 变换后的结果, 而 $\mu(t) = \mu(\tau)$, 那么拉长后的损失量

曲线应该比原曲线更平缓,如图 3 所示。

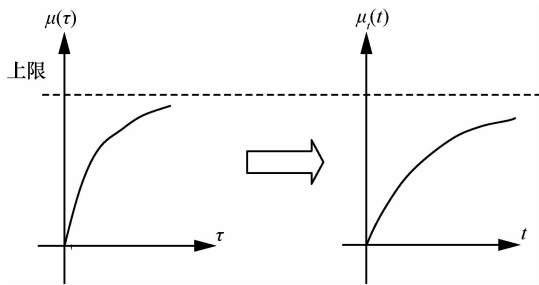


图 3 坐标系的变换

可以得出:

$$m(t)=\alpha(1-e^{-\beta t}) \tag{13}$$

损失密度函数为:

$$\lambda(t)=m'(t)=\alpha\beta e^{-\beta t} \tag{14}$$

同理,可以得到似然函数:

$$L(\alpha,\beta|\tau)=n\ln\alpha+n\ln\beta-\alpha(1-e^{-\beta\tau})+\beta\sum_{k=1}^n t_k \tag{15}$$

分别对 α,β,ω 求导,并对导数赋 0 建立方程组:

$$\begin{cases} \frac{n}{\alpha} = 1 - e^{-\beta(\tau_n + \omega\tau'_n)} \\ \frac{n}{\beta} = \alpha(\tau_n + \omega\tau'_n)s_n e^{-\beta(\tau_n + \omega\tau'_n)} + \sum_{k=1}^n (\tau_k + \omega\tau'_k) \\ 0 = -\alpha\beta X_n e^{-\beta(\tau_n + \omega\tau'_n)} + \beta\sum_{k=1}^n \tau'_k \end{cases} \tag{16}$$

用数值方法可求解出 α,β,ω 。

5 实际数据检验

使用与上文相同的数据和判断预测是否成功的标准,运用提出的改进模型,计算结果见表 4($\alpha=190,\beta=0.003\ 4,\omega=76.677\ 8$,准确率约为 50%)、表 5($\alpha=181,\beta=0.003\ 1,\omega=49.950\ 1$,准确率约为 62%)、表 6($\alpha=155,\beta=0.003\ 8,\omega=51.475\ 6$,准确率约为 50%)。

6 结束语

以上判断准确率的标准是可以变动的,实际应用中未必选择±0.5 作为标准。由于本文把损失分为 4 个等级,严

表 4 Windows XP 改进模型预测结果

序号	损失/万元	天数/日	累计损失/万元	估计值/万元	是否准确
1	3	1 327	128	128.344 0	是
2	3	1 328	131	131.702 1	是
3	2	1 363	133	134.526 5	否
4	3	1 371	136	135.497 6	是
5	1	1 385	137	137.919 2	否
6	1	1 407	138	138.091 6	是
7	2	1 442	140	140.522 7	是
8	2	1 448	142	141.280 2	否
9	2	1 449	144	141.881 9	否
10	4	1 457	148	142.943 3	否

表 5 Flash Player 改进模型预测结果

序号	损失/万元	天数/日	累计损失/万元	估计值/万元	是否准确
1	3	1 327	128	128.344 0	是
2	3	1 328	131	131.702 1	是
3	2	1 363	133	134.526 5	否
4	3	1 371	136	135.497 6	是
5	1	1 385	137	137.919 2	否
6	1	1 407	138	138.091 6	是
7	2	1 442	140	140.522 7	是
8	2	1 448	142	141.280 2	否
9	2	1 449	144	141.881 9	否
10	4	1 457	148	142.943 3	否



表 6 Firefox 改进模型预测结果

序号	损失/万元	天数/日	累计损失/万元	估计值/万元	是否准确
1	3	728	103	103.162 8	是
2	2	729	105	105.886 4	否
3	2	745	107	106.368 8	是
4	2	766	109	108.505 8	是
5	2	770	111	110.974 8	是
6	2	820	113	111.754 9	否
7	2	826	115	114.346 3	是
8	3	828	118	115.107 2	否
9	2	832	120	116.037 0	否
10	2	835	122	118.885 8	否

格来讲即使是猜测,也有 25% 的准确率,所以本文提出的基本模型依然比猜测可靠。

从另一方面来讲,本文提出的改进模型把时间进行一个变换,即拉长时间的目的为了使预测结果更接近真实数据,这包含两个方面的意义:第一,预测损失量的趋势应该与实际数据相同;第二,预测损失量的曲线应该与实际数据吻合得比较好。那么,就画出两个模型的结果和实际数据的曲线,如图 4 所示。

可见,预测曲线的趋势与实际是一致的,3 组数据中,改进模型的曲线明显比原始模型的曲线更贴近实际曲线,这也和表格中的数据吻合。通过拉长时间轴,得到了更好的预测结果。

参考文献:

- [1] SCHULTZ J E, BROWN D S, LONGSTAFF T A. Responding to computer security incidents[EB/OL]. (1990-07-23)[2015-09-20]. <ftp://ftp.cert.dfn.de/pub/docs/csir/ihg.ps.gz>.
- [2] PFLEEGER C P. Security in computing [M]. Upper Saddle River: Prentice-Hall, 1997: 46-48.
- [3] SHIN Y, WILLIAMS L. Is complexity really the enemy of software security [C]//The 4th ACM Workshop on Quality of Protection, October 27-31, Alexandria, VA, USA. New York: ACM Press, 2008: 47-50.
- [4] ANDERSON R. Security in open VeTSUS closed systems-the dance of boltzmann, coase and moore [C]//The Conference on Open Source Software Economics, Jul 9, 2002, London, UK. Cambridge: MIT Press, 2002: 1-15.
- [5] MUSA J D, IANNINO A, OKUMOTO K. Software reliability engineering[M]. New York: McGraw-Hill, 1999: 193-223.
- [6] MUSA J D, OKUMOTO K. A logarithmic Poisson execution time model for software reliability measurement [C]// The 7th Int'l Conference on Software Engineering. Orlando: IEEE Press, 1984: 230-238.
- [7] RESCORLA E. Is fining security holes a good idea [J]. IEEE Security&Privacy, 2005, 3(1): 14-19.
- [8] BECKER S, HASSELBRING W, PAUL A, et al. Trustworthy software systems: a discussion of basic concept and terminology[J]. ACM Sigsoft Software Engineering Notes, 2006, 31(6): 1-18.
- [9] 杨光宇, 曾东方, 罗平. 考虑短板效应的一种度量模型及其在软件可信性中的应用 [J]. 计算机应用研究, 2012(1): 165-167.

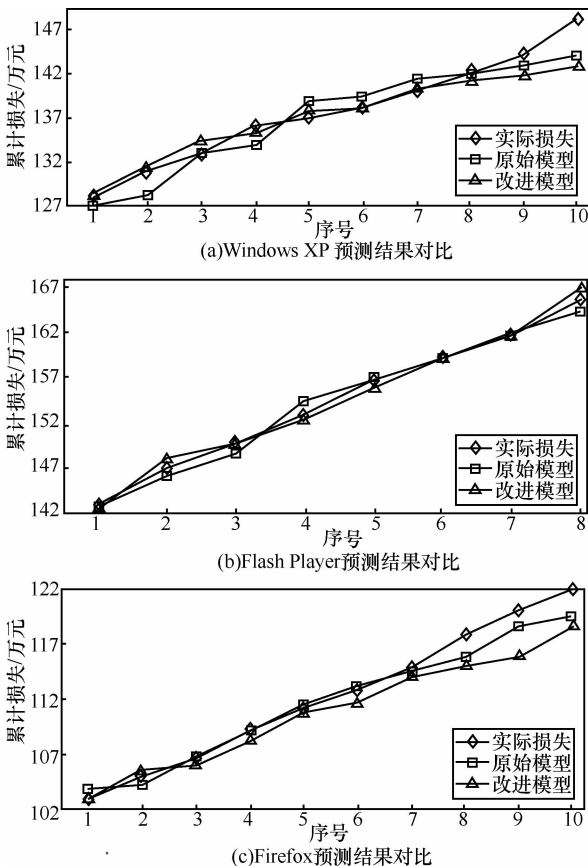


图 4 原始模型和改进模型曲线比较

- YANG G Y, ZENG D F, LUO P. Metric model considering effect of short board and its application in software trustworthiness[J]. Application Research of Computers, 2012(1): 165-167.
- [10] 王怀民, 刘旭东, 郎波, 等. 软件可信分级规范 v2.0[R/OL]. [2009-05-30]. <http://www.doc88.com/p-3008711993507.html>.
- WANG H M, LIU X D, LANG B, et al. Software trustworthiness classification specification v2.0 [R/OL]. [2009-05-30]. <http://www.doc88.com/p-3008711993507.html>.
- [11] 王怀民, 唐扬斌, 尹刚, 等. 互联网软件的可信机理[J]. 中国科学 E 辑, 2006, 36(10): 1156-1169.
- WANG H M, TANG Y B, YIN G, et al. The trusted mechanism of Internet software [J]. Science in China (E), 2006, 36(10): 1156-1169.
- [12] LIU Y Z, ZHANG L, LUO P, et al. Research of trustworthy software system in the network [C]// The 5th International Symposium on Parallel Architectures, Algorithms and Programming, Dec 17-20, 2012, Taipei, China. New Jersey: IEEE Press, 2012: 287-294.
- [13] VOAS J. Why is it so hard to predict software system trustworthiness from software component trustworthiness[C]// The 20th IEEE Symposium on Reliable Distributed Systems, October 28-31, 2001, New Orleans, Louisiana, USA. New Jersey: IEEE Press, 2001:179.
- [14] MUSA J D. A theory of software reliability and its application[J]. IEEE Transactions on Software Engineering, Los Alamitos, 1975, 1(3): 312-372.
- [15] MUSA J D, OKUMOTO K. A logarithmic Poission excution time model for software reliability measurement [C]// The 7th International Conference on Software Engineering. [S.l.]: Whippany Bell Laboratories, 1984: 230-238.

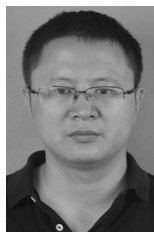
[作者简介]



彭轶(1984-),男,全球能源互联网研究院信息安全工程师,主要研究方向为网络与信息安全、数据库安全。



郭昊(1981-),男,全球能源互联网研究院信息安全工程师,主要研究方向为网络与信息安全、数据库安全。



王涛(1977-),男,全球能源互联网研究院信息安全工程师,主要研究方向为网络与信息安全、密码算法实现及应用。