一种基于GA-FAHP的软件漏洞风险评估方法

唐成华[1,3] 田吉龙[1,3] 汤申生[2] 张 鑫[3] 王 璐[3]
(桂林电子科技大学广西信息科学实验中心 桂林 541004)[1]
(西密苏里州立大学电子工程学院 圣约瑟夫 64507)[2]
(桂林电子科技大学广西可信软件重点实验室 桂林 541004)[3]

摘 要 针对软件系统中漏洞的风险等级确定等问题,提出了一种利用遗传模糊层次分析法(GA-FAHP)评估软件漏洞风险的方法。该方法首先利用改进的模糊层次分析法求出各风险因素权重,并建立模糊判断矩阵;其次将模糊判断矩阵的一致性检验与修正计算过程转化为带约束的非线性系统优化问题,并利用遗传算法求解;最后,通过GA-FAHP算法求出软件漏洞的风险值。实验结果表明,该方法具有较好的准确性和有效性,为软件漏洞风险评估提供了一种可行的途径。

关键词 软件漏洞,风险评估,遗传算法,模糊判断矩阵
中图法分类号 TP309 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.9.025

Risk Assessment of Software Vulnerability Based on GA-FAHP

TANG Cheng-hua[1,3] TIAN Ji-long[1,3] TANG Shen-sheng[2] ZHANG Xin[3] WANG Lu[3]
(Guangxi Experiment Center of Information Science, Guilin University of Electronic Technology, Guilin 541004, China)[1]
(Department of Engineering Technology, Missouri Western State University, St. Joseph MO 64507, USA)[2]
(Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China)[3]

Abstract Aiming at the problem of the vulnerability risk level determination in the software system, a genetic fuzzy analytic hierarchy process(GA-FAHP) approach was proposed to evaluate the risk of software vulnerability. Firstly, the improved FAHP is used to calculate the weight of each risk factor, and the fuzzy judgment matrix are established. Secondly, the consistency checking and correcting process of the fuzzy judgment matrix are transformed into an optimization problem for nonlinear constrained system, and the genetic algorithm is used to solve it. Finally, the risk degree of the vulnerability is calculated by GA-FAHP algorithm. Experimental results show that this method has good accuracy and validity, and provides a feasible way for the software vulnerability risk assessment.

Keywords Software vulnerability, Risk assessment, Genetic algorithm, Fuzzy judgment matrix

软件漏洞的普遍性及其后果的严重性是导致信息安全问题的根源之一。由于软件开发技术的缺陷、程序逻辑问题和结构漏洞等原因,需要通过代码重现、静态分析、软件扫描等方式对软件漏洞进行有效的风险调查、评估等管理过程[1]。漏洞管理在保障安全配置和阻止攻击中发挥着重要作用,是当前信息系统安全研究的重要内容[2]。作为系统安全管理的重要环节,软件漏洞风险评估可以为安全策略提供重要的参考价值,软件漏洞评估的模型与算法研究已日益受到重视。

目前,CVE(Common Vulnerabilities & Exposures)已整合了美国国家漏洞库NVD(National Vulnerability Database),但仍由NVD发布CVE并保持数据的更新。NVD包含了广泛的漏洞划分,可以用来挖掘对网络攻击的效果[3]。CVE的目的是使不同的漏洞数据库能够相互兼容,并能关联和共享同一漏洞信息,但它所描述的漏洞分类层次较低,并不能满足漏洞分类标准的需要[4]。由美国国家基础设施顾问委员会NIAC(National Infrastructure Advisory Committee)开发的CVSS(Common Vulnerability Scoring System)是一个行业公开标准,可用来评测漏洞的严重程度,但它定位于通用的漏洞评估方法,针对特定的漏洞尚不能够准确有效地进行评估[5]。文献[6]提出了一种针对主机安全性的量化融合模型,其通过对目标主机安全信息的信任度融合及关联,对目标主机进行漏洞存在可能性分析及漏洞的可利用性分析,但该模型没有给出漏洞风险的高低,也没有给出如何确定漏洞可利用性影响因素,以及安全性量化指标之间的转换规则。文献

到稿日期:2014-09-12 返修日期:2014-11-24 本文受国家自然科学基金(61462020,61363006,61163057),广西自然科学基金(2014GXNSFAA118375),广西信息科学实验中心基金(20130329),桂林电子科技大学研究生教育创新计划项目(GDYCSZ201421),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,主要研究方向为网络与信息安全,E-mail:tch@guet.edu.cn;田吉龙(1989—),男,硕士生,主要研究方向为网络信息安全;汤申生(1969—),男,博士,主要研究方向为智能信息处理;张 鑫(1990—),男,硕士生,主要研究方向为网络信息安全;王 璐(1991—),女,硕士生,主要研究方向为网络信息安全。

[7]引入路径风险与主体风险的概念，提出了一种面向漏洞关联网络的漏洞风险评估的定量方法与实现步骤。文献[8]主要分析了漏洞可利用性影响因素并对其进行量化处理，并通过运用传统的层次分析法获取各因素权重，利用灰色系统理论实现对可利用性因素的量化，计算过程复杂，很难满足系统实时性要求。文献[9]建立了主机安全威胁模型，计算漏洞对系统安全威胁的影响，提出一种基于主机访问图的漏洞严重程度的评估模型，其计算是基于经验实现对系统访问关系和主机重要度的参数量化，也没有考虑各漏洞是否被利用成功。文献[10]针对系统漏洞的严重性程度问题，以专家评定的方式确定风险因素权重，提出一种具有可操作性的基于层次分析法的系统漏洞量化评估方法。文献[11]提出基于 CVE 漏洞库的生存性量化分析数据库和量化算法，利用 CVE 漏洞库的漏洞检索项和工具包对被测系统进行模拟攻击，利用攻击结果计算系统的生存性量化值，该研究没有给出确定各属性的影响权重值的算法。IBM 公司的 ISS X-Force 采用定性的漏洞评估方法，给出定性的评价结果，并且主要是从攻击效果上来区分的，因此，对漏洞的风险因素考虑较少，不能真实反映漏洞的风险级别。文献[12]采用定性与定量相结合的方法，基于 CVSS 定量评分，最后给出一个漏洞评价分值，并没有考虑各风险因素权重，忽略了漏洞发生的概率，如果漏洞本身很难被触发，其危险级别再高，也不具有实际的危险性。

实际上，同一个漏洞在不同的软件或系统中被触发的条件是不一样的，即同一个漏洞在不同的环境被触发的概率是不同的。因此，对软件漏洞进行风险评估，应针对具体环境和特定对象，综合考虑全面因素。

定义 1 漏洞风险是指漏洞的多个影响因素相互作用而产生的危险性。

在定义 1 的基础上，本文提出一种基于遗传模糊层次分析法(GA-FAHP)的软件漏洞风险评估方法，从系统工程的角度，充分考虑漏洞各风险因素，利用模糊层次分析法构建软件漏洞评估指标体系，采用遗传算法求解模糊判断矩阵的一致性问题，通过综合评价计算漏洞风险值，判断出软件漏洞的优先级。该方法针对性强、简单实用且准确有效，为软件漏洞风险判别提供了有益探索。

1 软件漏洞风险评估算法

由于模糊层次分析法在验证模糊判断矩阵一致性问题时计算复杂且缺乏准确度，因此基于遗传算法对模糊层次分析法进行改进，将模糊层次分析法的一致性验证转化为最优解问题，从而简化计算步骤，提高验证效率。

1.1 改进模糊层次分析法构建模糊判断矩阵

虽然模糊层次分析是一种定性与定量相结合的决策分析方法，并在多目标、多准则的非结构化决策中得到广泛应用，但它不仅在检验模糊判断矩阵一致性时比较复杂，而且当判断矩阵不具有一致性时，对模糊判断矩阵的修正调节需要重复多次才能满足一致性，尤其是此时并不能保证最优。通常采用枚举法、启发式算法求最优解或近似最优解，但当枚举空间比较大时，求解效率较低，或者是需要特有的启发式规则来求解特殊的问题，缺少通用性。因此，本文利用遗传算法来弥补这些不足，通过优化算法对其进行改进，将模糊判断矩阵的一致性检验转化成非线性优化问题求解。

模糊层次分析法在构建模糊判断矩阵时采用 0-1 标度法，其模糊标度的含义如表 1 所列 。

表 1 模糊标度的含义

标度值	含义
0.5	两个元素相比，它们具有同等重要性
0.6	两个元素相比，一个元素比另一个元素稍重要
0.7	两个元素相比，一个元素比另一个元素明显重要
0.8	两个元素相比，一个元素比另一个元素比较重要
0.9	两个元素相比，一个元素比另一个元素极端重要
0.55、0.65、0.75、0.85	成对元素的差别介于两者之间，可取相邻判断的中间值
互补数	若元素 i 与元素 j 重要性之比为 b_{ij}，那么元素 j 与元素 i 的重要性之比为 $b_{ji}=1-b_{ij}$

通过专家两两比较软件漏洞的评价指标体系中同一层次的各指标重要性，根据表 1 实现量化，再对各个专家的数字标度进行加权平均，即得到专家的模糊判断矩阵 B：

$$B=\begin{bmatrix} 0.5 & b_{12} & \cdots & b_{1n} \\ b_{21} & 0.5 & \cdots & b_{2n} \\ \vdots & \vdots & \vdots & b_{3n} \\ b_{n1} & \cdots & b_{n(n-1)} & 0.5 \end{bmatrix} \tag{1}$$

其中，元素 $b_{ij}(i,j=1,2,\cdots,n)$表示第 i 个因素的重要性与第 j 个元素的重要性之比。

利用方根法求权重的步骤如下：

1)求矩阵各行的乘积的方根

$$S(i)=\left(\prod_{j=1}^{n} b_{ij}\right)^{\frac{1}{n}} \tag{2}$$

2)归一化处理

$$\overline{S}(i)=\frac{S(i)}{\sum_{i}^{n} S(i)} \tag{3}$$

3)组成权向量

$$w=(\overline{S}(1),\overline{S}(2),\cdots,\overline{S}(n)) \tag{4}$$

1.2 模糊判断矩阵的一致性

模糊判断矩阵的一致性反映了人们思维判断的一致性。但在实际决策分析中，由于所研究问题的复杂性和人们认识上可能的片面性，使构造出的判断矩阵往往缺少一致性。因此，要使权重结果准确可靠，必须进行一致性检验。

设漏洞的影响因素为 $a_1,a_2,\cdots,a_n$，其权重分别为 $w_1,w_2,\cdots,w_n$。由 b_{ij} 的定义知，b_{ij} 表示元素 a_i 比元素 a_j 重要的隶属度，b_{ij} 越大，a_i 就比 a_j 越重要，$b_{ij}=0.5$ 时，表示 a_i 和 a_j 同等重要。另一方面，由权重的定义知，w_i 是对元素 a_i 的重要程度的一种度量，w_i 越大，元素 a_i 就越重要。因此，w_i-w_j 的大小在一定程度上也表示了元素 a_i 比 a_j 重要的程度，且 w_i-w_j 越大，a_i 比 a_j 就越重要。这样，通过两两比较得到的元素 a_i 比 a_j 重要的重要程度度量 b_{ij} 与 w_i-w_j 满足如下关系：

$$b_{ij}=0.5+c(w_i-w_j),i,j=1,2,\cdots,n \tag{5}$$

其中，c 是两元素差异程度的一种度量。由于 c 的取值与评价元素的个数有关，因此一般取 c 值为$(n-1)/2$。

如果模糊判断矩阵满足式(5)，并能精确度量 $c(w_i-w_j)$，则此时模糊判断矩阵 B 具有完全一致性，于是有：

$$\lim\left(\sum_{i}^{n}\sum_{j}^{n}|0.5+c(w_i-w_j)-b_{ij}|\right)=0 \tag{6}$$

对软件漏洞的各影响因素及其后果严重性的掌握程度决

定了判断矩阵 B 一致性程度的高低，一般来说它们成正比。当式(6)严格成立时，判断矩阵 B 具有完全一致性。因此，其一致性检验问题可归纳为下列优化问题：

$$\begin{cases} \min F_{CI}(r_{ij},w_i,w_j)=\dfrac{\sum\limits_{i=1}^{n}\sqrt{\dfrac{1}{n}\sum\limits_{j=1}^{n}(r_{ij}-\overline{r_{ij}})^2}}{n}+ \\ \lim\dfrac{\sum\limits_{i}^{n}\sum\limits_{j}^{n}|0.5+c(w_i-w_j)-b_{ij}|}{n} \\ \text{s.t.}\begin{cases}0<w_i<1,0<w_j<1\\ \sum\limits_{i=1}^{n}w_i=1,\sum\limits_{j=1}^{n}w_j=1\end{cases} \end{cases} \tag{7}$$

其中，$F_{CI}(r_{ij},w_i,w_j)$ 为一致性指标函数。模糊判断矩阵具有一致性的前提是任意指定行和其余各行对应元素之差为某一个常数。因此，为了不失一般性，假设专家对第一行元素最有把握，进而根据这些有把握的元素对模糊判断矩阵进行一致性修正，于是，可设 r_{ij} 是第一行与其余各行对应元素之差，$\overline{r_{ij}}$ 是第一行与其余各行对应元素之差的平均值，从而通过计算标准差来反映模糊判断矩阵的一致性。式(7)是一个复杂的非线性函数优化问题，采用遗传算法，通过一系列选择、交叉和变异等遗传操作，最终得到该问题的全局最优解。显然，F_{CI} 值越小，则模糊判断矩阵 B 的一致性就越高，当取全局最小值 $F_{CI}=0$ 时，模糊判断矩阵 B 具有完全一致性。但在实际中一般以 0.1 作为分界线，即当进化搜索判断矩阵 B 的一致性指标函数值 F_{CI} 小于 0.1 时，认为判断矩阵 B 具有满意的一致性，据此计算的各因素的单排序权值 w_i 是可以被接受的。而当进化搜索判断矩阵 B 的一致性指标函数值 F_{CI} 大于 0.1 时，则调整判断矩阵 B，直至满足一致性。

1.3 改进的模糊层次一致性验证

遗传算法通过随机产生待求问题的初始解，形成初始种群，每个初始解即为种群个体，再通过遗传操作来达到种群进化的目的，即通过计算种群个体适应度来进行优秀个体选择、种群个体间交叉以及种群个体本身变异等操作来产生新一代个体，通过对种群个体逐代进行这些遗传操作，算法终止于收敛阈值或最大进化代数。

由于对全局最优或全局近似最优解具有较强的搜索能力，因此可以利用遗传算法解决模糊层次分析法在一致性验证求解时复杂繁琐又不能保证完全正确的问题。首先确定目标函数和约束条件，即转化为非线性优化问题；其次将目标的倒数作为遗传算法中的适应度函数，计算新种群个体的适应度，从中选择优秀个体来进行交叉和变异操作，如此迭代，当前后两代适应度之差小于 0.001 时算法终止，否则调整模糊判断矩阵继续迭代，直到满足终止条件。

算法 1 模糊判断矩阵一致性算法

```
Input:模糊判断矩阵 B,模糊判断矩阵的列数 s 和行数 m,初始种群优秀个体 j,种群个体数 N,种群 Population[N],两个种群个体间发生交叉的概率 CrossoverProbability,种群个体发生变异的概率 MutationProbability
Output:最小值 MinFCI
Begin {
1. for k=1 to s do
2.    Product=MatrixEachRowProduct(B);
      /*计算模糊判断矩阵每行的乘积*/
3. end for
4. for a=1 to m do
      Sum+=Product[a];
5. end for
6. for c=1 to m do
      Weight[c]=Product[c]/Sum;/*计算权重*/
7. end for
8. for i=0 to (N-1) do
9.    Population[i].fitness=CalculateFitnees(Population[i]);
      /*本算法将目标函数的倒数即 1/minFCI(rij,wi,wj)作为个体适应度函数,得到个体适应度;Population[N].fitness 为所有种群个体的适应度*/
10. end for
11. EliteMembers=Selection(Population[N].fitness);
    /*选择本代优秀个体*/
12. while(j<EliteMembers.Size()) do
13.    if(random() < CrossoverProbability) do
       /*当 random()小于交叉概率时进行交叉*/
14.       NewPopulation=Corssover(EliteMembers[j],EliteMembers[j+1]);
          /*产生新一代个体*/
15.    end if
16.    j=j+2;
17. end while
18. for i=0 to NewPopulation.Size()-1 do
19.    if(random() < MutationProbability) do
       /*random()小于交叉概率时进行变异*/
20.       Mutation(NewPopulation);
          /*种群个体发生变异*/
21.    end if
22. end for
23. Populaiton=NewPopulation;
24. BetweenValue = GetBetweenValue(Populaiton[n-1].fitness,
    Population[n].fitness); /*取得第 n 个和 n-1 个适应度之差*/
25. if(BetweenValue < 0.001)
26.    MinFCI=1/GETMaxFitnessValue(Population[N].fitness);
       /*求得最优解 MinFCI*/
27. else
28.    AdjustmentMatrix(B);/*调整模糊判断矩阵*/
29.    ReturnsProgramStart();
30. end if }
End
```

1.4 GA-FAHP 遗传模糊层次分析法

通过算法 1 实现了模糊判断矩阵的一致性，即所求得的权重是可以采纳的，并且计算简单、准确性高。由于传统的模糊层次分析法在计算综合度量值时是递层向上汇聚成综合度量值，很难实时反映目标层相对因素层的变换，缺乏灵活性，因此有必要对最后评估量化结果进行改进，例如采用第 II，III 层的权向量和目标漏洞相对于第 III 层的权重向量之积作为综合度量值，使得最后结果更加可靠。

算法 2 GA-FAHP 算法

```
Input:第 II、III 层的权向量 W2[T]和 W3[P],目标漏洞相对于第 III 层的权重向量 Wq[Z],权向量 W2、W3 和 Wq 中的元素个数 T、P 和 Z,模糊判断矩阵一致性检验结果 FCI
Output:综合评价结果 S
Begin {
1. if(FCI<0.1)
2.    for t=1 to T do
3.       for p=1 to P do
4.          RightVectorProduct=W2[t]·W3[p];
```

```
        /* 计算第 II,III 层的权向量的积 */
5.      end for
6.  end for
7.  for z=1 to Z do
8.    S=RightVectorProduct · W_q[z];
      /* 求出最后的综合度量值 */
9.    end for
10. end if }
End
```

2 基于 GA-FAHP 的软件漏洞风险评估应用

2.1 构建安全风险评估指标体系

目前软件漏洞评估要素的分类组织与方法主要有：通用漏洞披露 CVE(common vulnerabilities and exposures)、常见漏洞列表 CWE(common weakness enumeration)、通用攻击模式列表和分类 CAPEC(common attack pattern enumeration and classification)、通用配置列举 CCE(common configuration enumeration)、通用平台列 CPE(common platform enumeration)、最可信的源计算机安全培训、认证和研究 SANS(the most trusted source for computer security training, certification and research)、开放 Web 应用安全项目 OWASP(open web application security project)、CWE/SANS 的前 25 名最危险的编程错误等[13]。从中总结出 10 个主要的评估要素，考虑 CVE、NVD、Microsoft[14]、Vupen[15]、US-CERT[16]、X-Force[17]等组织在漏洞风险评估中所采用的安全漏洞影响因素，如表 2 所列。

表 2 软件漏洞评估要素使用情况

影响因素	CVE	NVD	Microsoft	Vupen	US-CERT	X-Force
访问途径	√	√	√	√	√	√
访问复杂度	√	√		√	√	√
授权认证	√	√		√	√	√
完整性影响	√	√	√	√	√	
可用性及影响	√	√	√	√	√	√
机密性影响	√	√		√	√	√
修补等级	√	√	√		√	
间接破坏风险	√	√				
目标分布	√	√			√	
报告可信度	√	√				√

在这 10 种影响因素中，出现次数≥4 的因素共 7 个：访问途径、访问复杂度、授权认证、完整性影响、可用性及影响、机密性影响和修补等级，由此可以说明这 7 个属性是普遍承认的安全漏洞中具有代表性的因素。因此结合本文漏洞风险评估的实际需要，还添加了一些其他相关因素，并将这些因素划分成 4 个组：危险程度 A、风险概率 B、利用复杂度 C 和修补难度 D，如图 1 所示。

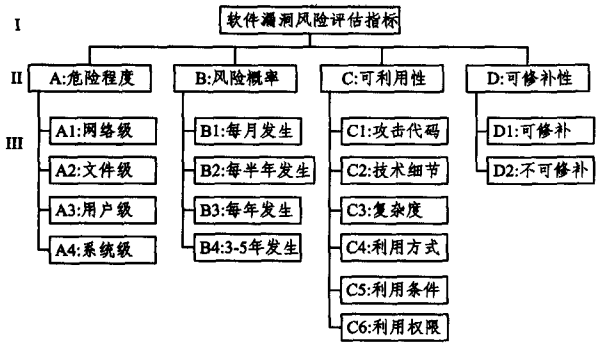

图 1 软件漏洞风险评估指标体系

2.2 安全风险评估指标体系流程

模糊层次分析法是一种定性与定量评价相结合的决策分析方法，其核心是保证模糊判断矩阵的一致性。但是由于许多不确定性因素的影响，需要经过多次调整才能满足一致性要求，计算量很大，特别是在因素众多且各因子的相对重要性随分析过程不断变化的情况下，该问题显得尤为突出。

为了解决软件漏洞评估过程中判断矩阵的一致性问题，可以将验证一致性问题转化为有约束的非线性优化问题，通过遗传算法进行求解，最终获得主要因子及其权值。软件漏洞的安全风险评估流程如图 2 所示。

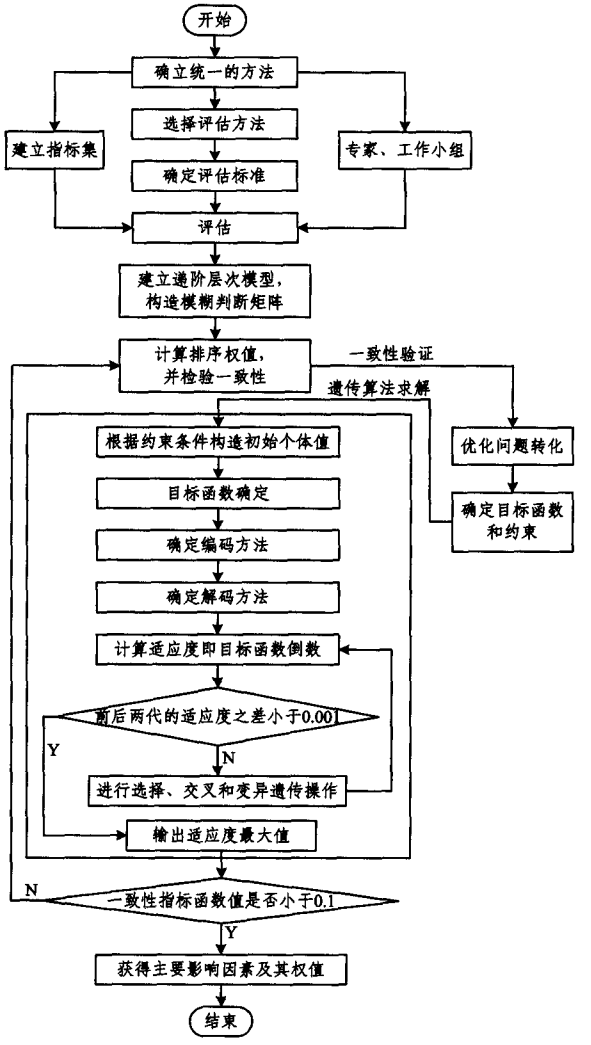

图 2 软件漏洞风险评估流程

3 实验结果及分析

3.1 构建安全风险评估指标体系

以 OpenSSL 中的漏洞 CVE-2014-0160、CVE-2014-0195 和 CVE-2014-0198 为例，将其分别表示为 X_1、X_2 和 X_3，通过查询 CVE、CNNVD[18] 和 NVD 3 个漏洞数据库给出的相关定量指标数据，对定性指标进行数据收集，并通过专家分析对数据进行标准化处理后得到各层次的模糊判断矩阵。根据 GA-FAHP 算法，利用 Matlab 7.8 求解各层次下的评价指标权重值。第 II 层危险程度、风险概率、可利用性、可修补性的模糊判断矩阵及评价指标权重值如表 3 所列，第 III 层各影响因素的模糊判断矩阵及评价指标权重值分别如表 4—表 7 所列。

表 3　第 II 层模糊判断矩阵及权重值

II 层	A	B	C	D	W_2
A	0.5	0.6	0.7	0.8	0.4325
B	0.4	0.5	0.4	0.7	0.2273
C	0.3	0.6	0.5	0.6	0.2246
D	0.2	0.3	0.4	0.5	0.1156

表 4　漏洞危险程度模糊判断矩阵及权重值

A	A1	A2	A3	A4	W_A
A1	0.5	0.4	0.3	0.1	0.1479
A2	0.6	0.5	0.4	0.2	0.2092
A3	0.7	0.6	0.5	0.3	0.2663
A4	0.9	0.8	0.7	0.5	0.3766

表 5　漏洞风险概率模糊判断矩阵及权重值

B	B1	B2	B3	B4	W_B
B1	0.5	0.6	0.7	0.9	0.4853
B2	0.4	0.5	0.7	0.8	0.3043
B3	0.3	0.3	0.5	0.6	0.1346
B4	0.1	0.2	0.4	0.5	0.0758

表 6　漏洞可利用性模糊判断矩阵及权重值

C	C1	C2	C3	C4	C5	C6	W_C
C1	0.5	0.4	0.3	0.6	0.4	0.2	0.1317
C2	0.6	0.5	0.3	0.4	0.3	0.1	0.1165
C3	0.7	0.7	0.5	0.8	0.9	0.3	0.2228
C4	0.4	0.6	0.2	0.5	0.6	0.1	0.1222
C5	0.6	0.7	0.1	0.4	0.5	0.4	0.1407
C6	0.8	0.9	0.7	0.9	0.6	0.5	0.2608

表 7　漏洞可修补性模糊判断矩阵及权重值

D	D1	D2	W_D
D1	0.5	0.9	0.75
D2	0.1	0.5	0.25

通过查询有关资料得到 X_1、X_2 和 X_3 3 个漏洞针对各评价指标的模糊判断矩阵，其权重计算结果如表 8 所列。

表 8　3 个漏洞相对于各评价指标的权重值

W_i	X_1	X_2	X_3
W_{A1}	0.3335	0.2647	0.4019
W_{A2}	0.2747	0.4316	0.2892
W_{A3}	0.4529	0.2368	0.3103
W_{A4}	0.2571	0.4858	0.2571
W_{B1}	0.3142	0.4783	0.2057
W_{B2}	0.3848	0.4803	0.1348
W_{B3}	0.1646	0.4041	0.4313
W_{B4}	0.4406	0.3728	0.1866
W_{C1}	0.2368	0.4529	0.3103
W_{C2}	0.4850	0.3322	0.1828
W_{C3}	0.1746	0.3677	0.4577
W_{C4}	0.3578	0.4282	0.2141
W_{C5}	0.4732	0.3710	0.1558
W_{C6}	0.3284	0.4554	0.2163
W_{D1}	0.2892	0.2747	0.4361
W_{D2}	0.3450	0.4158	0.2392

由表 8 数据，可以得到 $W_2=(0.4325, 0.2273, 0.2246, 0.1156)$，$W_3=(W_A, W_B, W_C, W_D)$，$W_q=(W_{A1}, W_{A2}, W_{A3}, W_{A4}, W_{B1}, W_{B2}, W_{B3}, W_{B4}, W_{C1}, W_{C2}, W_{C3}, W_{C4}, W_{C5}, W_{C6}, W_{D1}, W_{D2})$，根据算法 2，可以计算出 $S=(0.3558, 0.3650, 0.2744)$，因此 3 个漏洞中 X_2 最危险，需要重点防范。

基于 GA-FAHP 的软件漏洞风险评估的求解结果精确到小数点后 4 位，一般其他的漏洞评估方法精确到小数后 2 位，所以计算结果更加精确，更能准确可靠地比较漏洞的危险性，避免因漏洞评估过程中漏洞值相等或很相近无法比较出哪个漏洞更危险而必须重新调整参数重复计算的问题。

为了方便分析，图 3 给出了 3 个漏洞相对于第 III 层的权重变化趋势。

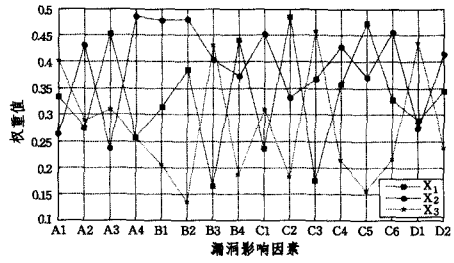

图 3　漏洞的权重变化趋势

从图 3 可以看出，X_2 曲线的绝大部分在其他漏洞曲线之上，因此判断出 X_2 相对更危险；X_1 在 B_3 处比其它两个的都低，即利用复杂度较低，因为 X_1 漏洞源于程序没有正确处理 Heartbeart Extension 数据包。远程攻击者可借助特制数据包利用该漏洞获取进程内存的敏感信息(如读取私钥)。

实验中还将 GA-FAPH 遗传模糊层次法与单纯的模糊层次分析法在求解一致性问题上的所用时间进行比较，采用 5 组数据进行对比，漏洞个数分别为 20,30,40,50,60，如图 4 所示。

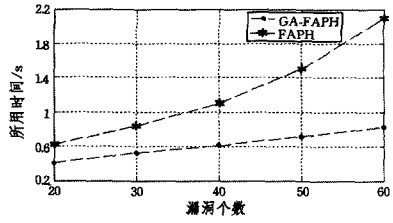

图 4　一致性检验所用时间对比

从图 4 可以看出，单纯的模糊层次分析法的曲线上升幅度随着漏洞数量的增加不断变大，即随着漏洞数量的增多，单纯的模糊层次分析法的效率下降明显；而 GA-FAPH 遗传模糊层次分析的曲线变化趋势很慢，表明采用 GA-FAPH 算法解决一致性验证问题更加有效、可靠。

结束语　本文提出了基于遗传模糊层次分析法的软件漏洞风险评估方法，将复杂繁琐的模糊一致性检验转化成非线性优化问题，并利用遗传算法求解，计算更加科学与准确。经过实验过程与分析表明，该方法能有效准确地评估软件漏洞的风险。不足之处在于，漏洞的评价体系还不能涵盖所有的因素，因此，下一步主要致力于改进漏洞影响因素的归类划分及评价方式。

参 考 文 献

[1] Sedaghat S, Adibniya F, Sarram M A. The investigation of vulnerability test in application software[C]// Proceeding of the 2009 International Conference on the Current Trends in Information Technology. 2009:1-5

[2] Martin B, Remi B, Olivier F. Vulnerability assessment in autonomic networks and services: a Survey[J]. IEEE Communications Surveys & Tutorials, 2014, 16(2):988-1004

[3] Jason L W, Jason W L, Miles A M. Estimating software vulnerabilities a case study based on the misclassification of bugs in MySQL server[C]// Proceeding of the 2013 Eighth International Conference on Availability, Reliability and Security. Regensburg, Germany, 2013:72-81

(下转第 158 页)

参 考 文 献

[1] Rothermel G, Harrold M J. Analyzing regression test selection techniques [J]. IEEE Transactions on Software Engineering, 1996, 22(8): 529-551

[2] Rothermel G, Untch R H, Chu C Y, et al. Prioritizing test cases for regression testing [J]. IEEE Transactions on Software Engineering, 2001, 27(10): 929-948

[3] Li Z, Harman M, Hierons R M. Search algorithms for regression test case prioritization [J]. IEEE Transactions on Software Engineering, 2007, 33(4): 225-237

[4] Srikanth H, Williams L, Osborne J. System test case prioritization of new and regression test cases[C] // 2005 International Symposium onEmpirical Software Engineering, 2005. IEEE, 2005: 10

[5] Elbaum S, Malishevsky A G, Rothermel G. Prioritizing testcases for regression testing [C] // Proceedings of the International Symposium on Software Testing and Analysis. 2000: 102-112

[6] Kim J M, Porter A. A history-based test prioritization technique for regression testing in resource constrained environments[C] // Proceedings of the 24rd International Conference on Software Engineering(ICSE 2002). IEEE, 2002: 119-129

[7] Qu Bo, Nie Chang-hai, Xu Bao-wen, et al. Test case prioritization for black box testing[C] // 31st Annual International Computer Software and Applications Conference, 2007(COMPSAC 2007). IEEE, 2007, 1: 465-474

[8] 屈波，聂长海，徐宝文. 基于测试用例设计信息的回归测试优先级算法[J]. 计算机学报，2008，31(3)：431-439
Qu Bo, Nie Chang-hai, Xu Bao-wen. Test case prioritization based on test suite design information [J]. Chinese Journal of Computers, 2008, 31(3): 431-439

[9] Walcott K R, Soffa M L, Kapfhammer G M, et al. Timeaware test suite prioritization[C] // Proceedings of the 2006 International Symposium on Software Testing and Analysis. ACM, 2006: 1-12

[10] Elbaum S, Rothermel G, Kanduri S, et al. Selecting a cost-effective test case prioritization technique[J]. Software Quality Journal, 2004, 12(3): 185-210

[11] Rothermel G, Untch R H, Chu C, et al. Prioritizing test cases for regression testing[J]. IEEE Transactions on Software Engineering, 2001, 27(10): 929-948

[12] 屈波，聂长海，徐宝文. 回归测试中测试用例优先级技术研究综述[J]. 计算机科学与探索，2009，3(3)：225-233
Qu Bo, Nie Chang-hai, Xu Bao-wen. Survey of Test Case Prioritization for Regression Testing [J]. Journal of Frontiers of Computer Science and Technology, 2009, 3(3): 225-233

(上接第 138 页)

[4] 陈波，师惠忠. 一种新型 Web 应用安全漏洞统一描述语言[J]. 小型微型计算机系统，2011，32(10)：1994-2001
Chen Bo, Shi Hui-zhong. Novel uniform vulnerability description language of Web application[J]. Journal of Chinese Computer System, 2011, 32(10): 1994-2001

[5] Jiang F, Dong Dao-yi, Cao Long-bing, et al. Agent-based self-adaptable context-aware network vulnerability assessment[J]. IEEE Transaction on Network and Service Management, 2013, 10(3): 255-270

[6] 陆余良，夏阳. 主机安全量化融合模型研究[J]. 计算机学报，2005，28(5)：914-920
Lu Yu-liang, Xia Yang. Research on target-computer secure quantitative fusion model [J]. Chinese Journal of Computers, 2005, 28(5): 914-920

[7] 周亮，李俊娥，陆天波，等. 信息系统漏洞风险定量评估模型研究[J]. 通信学报，2009，30(2)：71-76
Zhou Liang, Li Jun-e, Lu Tian-bo, et al. Research on quantitative assessment model on vulnerability risk for information system [J]. Journal of Communications, 2009, 30(2): 71-76

[8] 杨宏宇，朱丹，谢丽霞. 网络信息系统漏洞可利用性量化评估研究[J]. 清华大学学报(自然科学版)，2009，49(S2)：2157-2163
Yang Hong-yu, Zhu Dan, Xie Li-xia. Quantitative evaluation of vulnerability exploitability in network information systems[J]. Journal of Tsinghua University(Science and Technology), 2009, 49(S2): 2157-2163

[9] 宋舜宏，陆余良，杨国正，等. 一种应用主机访问图的网络漏洞评估模型[J]. 小型微型计算机系统，2011，32(3)：483-488
Song Shun-hong, Lu Yu-liang, Yang Guo-zheng, et al. Network vulnerability assessment model applying host-based access graphs [J]. Journal of Chinese Computer Systems, 2011, 32(3): 483-488

[10] 李鑫，李京春，郑雪峰，等. 一种基于层次分析法的信息系统漏洞量化评估方法[J]. 计算机科学，2012，39(7)：58-63
Li Xin, Li Jing-chun, Zheng Xue-feng, et al. Analytic hierarchy process (AHP)-based vulnerability quantitative assessment method for information systems[J]. Computer Science, 2012, 39(7): 58-63

[11] 王新喆，许榕生. 基于 CVE 漏洞库的生存性量化分析数据库和量化算法的设计[J]. 计算机应用，2008，28(2)：415-417，421
Wang Xin-zhe, Xu Rong-sheng. Design of survivability quantum analysis database and quantum algorithm based on CVE database[J]. Computer Applications, 2008, 28(2): 415-417, 421

[12] Liu Qi-xu, Zhang Yu-qing. VRSS: A new system for rating and scoring vulnerabilities[J]. Computer Communications, 2011, 34(3): 264-273

[13] Martin R A. Making security measurable and manageable[C] // Proceeding of the 2008 IEEE Military Communications Conference. San Diego, CA, 2008: 1-9

[14] Microsoft security response center security bulletin severity ratingsystem[EB/OL]. http://www.microsoft.com/technet/security/ bulletin/rating.mspx, 2012

[15] Vupen security [EB/OL]. http://www.vupen.com/english, 2012

[16] US-CERT. Vulnerability notes database field descriptions[EB/OL]. http://www.kb.cert.org/vuls/html/fieldhelp # metric, 2012

[17] IBM IIS X-Force[EB/OL]. http://xforce.iss.net, 2012

[18] China National Vulnerability Database of Information Security [DB/OL]. http://www.cnnvd.org.cn/vulnerability, 2014