软件脆弱性危险程度量化评估模型研究

李 艺[1] 李新明[2] 崔云飞[3]
(装备指挥技术学院科研部 北京 101416)[1] (装备指挥技术学院 EIES 重点实验室 北京 101416)[2]
(装备指挥技术学院研究生院 北京 101416)[3]

摘 要 软件脆弱性的危险程度是对软件脆弱性被利用来攻击系统的潜在危险的度量。在分析目前已知的相关评价方法及其局限性的基础上,提出了根据脆弱性影响的严重程度和脆弱性可利用性来评估脆弱性危险程度的分析框架,并基于模糊理论,提出了软件脆弱性危险程度评估的量化模型,建立了模糊测评因素关联隶属关系的递阶层次结构,并重点分析了基于模糊集的指标量化、基于模糊关系矩阵的指标权重的确定和软件脆弱性危险程度的综合评价方法。最后,给出了模型的应用与实现。

关键词 软件脆弱性,影响,危险程度,评估,模糊理论
中图法分类号 TP311 **文献标识码** A

Research of Evaluating Model on the Criticality of Software Vulnerability

LI Yi[1] LI Xin-ming[2] CUI Yun-fei[3]
(Department of Scientific Research, Institution of Command and Technology of Equipment, Beijing 101416, China)[1]
(EIES Key Laboratory, Institution of Command and Technology of Equipment, Beijing 101416, China)[2]
(Company of Postgraduate Management, Institution of Command and Technology of Equipment, Beijing 101416, China)[3]

Abstract The criticality of software vulnerability is the measurement of the potential risk of which the software vulnerability may be taken advantage for attacking the system. Based on analysis of current evaluation methods and their limitation, an analysis framework for evaluating the criticality of software vulnerability was proposed, according to the impact severity and probability of vulnerability. Based on fuzzy theory, the quantification model for evaluating the criticality of software vulnerability was proposed and the hierarchy of fuzzy evaluation factors' relationship and membership was established. The fuzzy set-based indices quantification, the fuzzy relational matrix-based indices weight value and the general evaluation method for software vulnerability criticality were emphasized. At last, the application and implement of the evaluating model were given.

Keywords Software vulnerability, Impact, Criticality, Evaluation, Fuzzy theory

软件脆弱性是指在软件的需求分析、设计、编码和运行阶段存在的漏洞,当软件运行在某个网络环境中,可以利用该漏洞危害系统的安全[1]。软件脆弱性最主要的特点是具有可利用性。软件脆弱性是导致绝大多数互联网上安全事件的根源[2]。只要有软件脆弱性存在,就一定存在一种或者多种利用该脆弱性产生危害性结果的方法。利用软件脆弱性可以破坏或者潜在威胁系统安全性的一个或者多个方面,包括系统的保密性、完整性、可用性、可控性和不可否认性等。评价软件脆弱性的危险程度,就是评价软件脆弱性对系统安全的危险性。

软件脆弱性的本质是可以被利用来破坏系统的安全,但每一个脆弱性对系统安全造成的影响并不一样,必须对软件脆弱性危险程度进行合理的评估。一般来说,评估有定性和定量两种分析方法。本文提出一种基于模糊理论的脆弱性危险程度的量化评估模型。

1 软件脆弱性危险程度评估方法研究现状

目前,一些安全组织和操作系统厂商定期发布一些软件脆弱性威胁程度的评估结果,为系统管理员在对系统进行维护时提供参考。这些评估大多都是定性评估,不太准确。美国基础设施顾问委员会于2004年提出通用缺陷评估系统CVSS(Common Vulnerability Scoring System)1.0版本,2007年修改后发布CVSS 2.0版本,通过对基本群、暂时群和环境群中的各种属性信息进行评分,最后得到一个综合的评分值。由于CVSS的各要素和计算公式等方面还存在许多问题,CVSS v2.0仍然没有得到微软等厂商的支持,漏洞评估整体上仍然是"各自为政"的局面[3]。

美国国家脆弱性库NVD、美国应急相应组US-CERT、美国安全研究机构SANS、法国安全研究组织FrSIRT、ISS公司的X-Force、微软公司、Symantec公司、Red Hat公司等对脆弱

到稿日期:2010-07-20 返修日期:2010-11-07 本文受863国家课题(2006AA01Z447)资助。
李 艺(1964—),女,教授,主要研究方向为操作系统和网络安全,E-mail:liyi1221@263.net;李新明(1964—),男,教授,主要研究方向为网络和系统集成;崔云飞(1986—),男,博士生,主要研究方向为信息系统。

性的严重等级都有一定程度的研究，都各有特点和可取之处，同时也存在各种缺点。比较典型的有 SANS(the consensus Security Alert NewSletter)和微软公司的危险程度评估方法。

SANS 将脆弱性的危险程度分为 4 个等级：危急、高、中、低[4]。其中“危急”级指脆弱性的利用能够破坏网络服务或能够使远程用户获得管理员权限，利用比较简单或者攻击代码已知，利用结果会破坏服务系统或高价值资产。如 samba 中 call_trans2open()函数存在的缓冲区溢出脆弱性；“高”级指脆弱性的利用不如危急级别的严重，受影响的软件或平台不是关键服务，利用一般比较困难，利用结果常常影响到客户端程序。如 Pam_smb 模块中存在的密码缓冲区溢出脆弱性；“中”级指脆弱性的利用对系统影响较小，或利用非常困难，或者需要较高的权限。如 apache 程序中内存耗尽引起的拒绝服务攻击；“低”级指脆弱性的利用对系统的影响很小，利用一般需要本地帐户或需要物理接触系统，利用结果常会影响客户端的机密性或造成拒绝服务或导致组织结构、系统配置、系统版本号、网络拓扑等信息的泄漏。

微软安全响应中心通过安全等级评估系统 SAS(Security Rating System)[5]对 Windows 系统中的每一个脆弱性的危险程度进行评估。评估结果分为 4 个等级：严重、重要、中等、低。“严重”级指能够使蠕虫病毒自动传播的脆弱性；“重要”级指能够危及用户数据的机密性、完整性、可用性或危及计算资源完整性、可用性的脆弱性；“中等”级指由于默认配置等因素使攻击难度减轻的脆弱性；“低”级指利用困难或对系统影响较小的脆弱性。

以上两种脆弱性等级评估都是对脆弱性危险程度定性的评估，没有公布具体的评估步骤和标准，缺乏相应的评估过程，不利于个人或组织的利用。

2 软件脆弱性危险程度评估框架

软件脆弱性的危险程度是对脆弱性被利用时对系统造成安全影响的严重程度的度量。危险程度首先与软件脆弱性造成影响的严重程度密切相关。但如果利用一个脆弱性需要特别复杂的条件，则即使利用该脆弱性可以造成特别严重的后果，该脆弱性的危险程度仍然可以认为较低。所以，软件脆弱性的危险程度还与脆弱性被利用的可能性密切相关。软件脆弱性的危险程度可以用公式表示为：

$$Criticality = Severity \times Probability$$

式中，Severity 为影响的严重程度，指脆弱性被最大利用时对系统安全造成的直接影响的严重程度，Probability 为可利用性，指软件脆弱性被最大利用的可能性。

2.1 软件脆弱性影响严重程度

脆弱性危险程度与脆弱性能造成影响的严重程度成正比，影响越严重的脆弱性，其危险程度越大。

脆弱性产生的影响是脆弱性的主要属性，许多脆弱性分类研究把产生的影响作为分类属性[6]，如 Wenliang Du 等人基于软件脆弱性从引入到消失的生命周期提出了一种三维的分类方法[7]，其中“破坏效果”一维分析了非法地执行代码等脆弱性影响；Matt Bishop 针对入侵检测系统提出了一个六维度的脆弱性分类法，其中“影响域”一维对脆弱性影响的范围进行了研究[8]；Krsul 提出了一个软件脆弱性影响的四维度分类法，从影响的对象、影响程度、实施影响的方法、攻击过程中的输入 4 个方面对脆弱性影响进行研究[9]；Power 等提出一种脆弱性所产生的威胁的分类法，将威胁分成威胁可用性和用途等类型；Perry 等提出基于攻击者和影响的二维分类方法，其中“影响”一维分成物理破坏等类型；一般脆弱性数据库中都有各自的脆弱性影响的分类方法，如普渡大学的 CERIAS 脆弱性数据库将影响分成访问数据、执行命令、执行代码、拒绝服务等 4 大类[10]。

上述的分类法目前没有一个被普遍接受，因为它们从不同的目的出发，都有各自的局限性，大多数都存在歧义性，很难去实施[11]，在确定性、完备性、适应性和扩展性方面各有优劣[12]。Wenliang Du 的分类法[7]没有给出直接影响的准确定义，也没有考虑影响的范围。Matt Bishop 分类法[8]主要集中在脆弱性影响的范围上，但对影响范围的描述并不全面。Krsul 的分类法[9]是最具体的，对影响的对象、实施影响的方法等进行了详尽的描述，但这种枚举方式在全面性、通用性、可扩展性等方面很难保证。CERIAS 脆弱性的分类方法[10]存在歧义性。

本文基于影响广度和深度来分析脆弱性产生的影响及其严重程度，影响的广度或范围从大到小可以分为系统级、用户级和文件级 3 个级别；在不同的影响广度之内，有不同的影响深度。

其中，系统级的影响是指脆弱性的影响是全局的，系统内所有实体都要受到影响或者影响到的对象关系到整个系统的安全。系统级的影响深度可划分成获得管理员的全部权限、更改任意文件、读任意文件、影响系统运行和泄漏系统信息等类型。这个级别的脆弱性有 wu-ftp 中存在的 CVE-1999-0081 脆弱性、Abyss Web Server 目录处理中存在的 CVE-2002-1079 脆弱性、Linux 路由缓冲中存在的 CVE-2003-0244 脆弱性、apache 中存在的 CVE-2004-0263 脆弱性等。

用户级的影响是指影响范围限定在某些用户能够访问的资源内，这里的用户指除拥有全部权限的 root 用户之外的所有用户。用户级影响深度可划分成获得用户全部权限、更改用户任意文件、读取用户任意文件、影响用户正常使用和泄漏用户信息等类型。这个级别的脆弱性有 Debian Linux 0.76 版本的 PAM 中存在 CVE-2002-1227 脆弱性。

文件级的影响是指脆弱性的影响范围只局限于某些特定的文件，此处的文件还包括目录、程序、系统函数库、Web 页、网络服务、网络连接、网络端口、网络数据包、设备等实体。文件级的影响深度可划分成改变文件执行状态、更改文件和读取文件等类型。这个级别的脆弱性如 wu-ftp 服务程序中的脆弱性 CVE-1999-0075、Redhat Linux 7.1 中 Xinetd 的 umask 默认值等于 0 的脆弱性等。

针对每一个脆弱性，首先确定脆弱性影响的广度，然后确定影响的深度，最后对脆弱性影响严重程度进行评估。

2.2 软件脆弱性可利用性

脆弱性危险程度与被利用可能性成正比，被利用的可能性越小的脆弱性，其危险程度越小。

文献[13]也把主机漏洞可利用性作为主机脆弱性量化评估的基础，列举了 12 条漏洞可利用性影响因素，如目标 IP 地址、漏洞发现时间、攻击该漏洞的技术成分等，可以看出在分类的粒度、完备性方面还有待完善。

与软件脆弱性可利用性密切相关的因素归纳成以下几

种：

1)默认安装情况。如果出现脆弱性的软件是默认安装的，则该脆弱性往往容易被系统管理员忽略，从而该脆弱性被发现、被利用的可能性较大。

2)稳定攻击代码情况。软件脆弱性最初一般是由一些技术较高的人发现和利用的，如果存在针对该脆弱性的稳定攻击代码，则被利用的可能性就较大。

3)攻击代码获得情况。尽管一些脆弱性的利用是很复杂的，但如果存在一些稳定的攻击代码，并容易获得，不具备相关技术的人也能成功地实施攻击，则脆弱性被利用的可能性较大。

4)利用复杂度情况。如果对某个脆弱性的利用不存在自动攻击程序，而且利用时要经过复杂的步骤或等待特定的时机，那么对该脆弱性成功利用的可能性就比较小。

5)技术细节获得情况。如果有关脆弱性的技术细节很容易得到，那么编写自动攻击代码的人数就会增加很多，从而利用脆弱性的可能性增大。

6)使用广泛性情况。广泛使用的软件中存在的脆弱性被利用的可能性是极大的，这一点已在实际中得到证明，如针对Windows系统的攻击代码随处可见。

7)利用权限情况。利用权限可以分成本地用户、远程用户和一般用户3种。本地用户指在目标机上拥有帐户并能物理访问目标机；远程用户指在目标机上拥有帐户但只能远程访问目标机的用户；一般用户指在目标机上没有帐户的用户，只能通过服务远程访问目标机。上述3种权限从大到小排列，利用该脆弱性需要的权限越大，该脆弱性被利用的可能性就越小。

8) 利用条件情况。利用脆弱性是否需要额外的条件，如是否需要与其它主机的协同操作，是否需要其它设备等，需要的利用条件越多，脆弱性被利用的可能性就越小。

9) 利用方式情况。脆弱性被利用的方式可以分为物理接触、主机模式、客户机模式、中间模式。物理接触是指攻击者需要物理地接触目标机才能利用该脆弱性；主机模式是一种最常见的脆弱性利用方式，攻击方为客户机，被攻击方为服务器；客户机模式与主机模式相反，攻击方为服务器，被攻击方为客户机，当用户访问网络上的服务器时，可能遭到服务器发送恶意命令的袭击；中间模式是指当攻击者位于被攻击方对外的通信线路上，可以观察或截获传输的信息。

脆弱性可利用性的评估要综合上述各种因素。软件脆弱性的可利用性越大，表示该脆弱性被利用的可能性越大，利用的条件越低、利用的方法越简单。反之，可利用性越低，表示该脆弱性被利用的可能性越小，利用的条件越不容易满足，利用的方法越困难。

3 基于模糊理论的软件脆弱性危险程度评估模型

本文基于模糊理论提出基于严重程度和可利用性的软件脆弱性危险程度评估模型。

3.1 模糊评测因素关联的递阶层次结构的建立

软件脆弱性危险程度评估的模糊测评因素关联隶属关系采用递阶层次结构，如图1所示。第一层为评估目标，即软件脆弱性危险程度，是整个综合评估的对象。第二层为评估指标，用来指定各个测评因素，采用分层结构，最底一层指标为测定级指标，不需要进一步分解，可直接给出量化值或做出直接评估。第三层为评估对象，即各软件脆弱性。

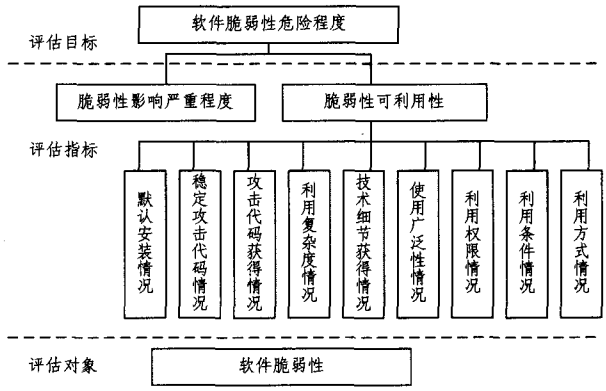

图1 软件脆弱性危险程度评估的递阶层次结构

3.2 基于模糊集的指标量化

评估的关键是基于模糊集确定测定级指标对各量化等级的隶属度并计算量化值。采用的方法是根据各测定级指标的性质，设定对应的调查表，然后通过向领域专家调查取证的方法，计算指标的量化值。需要量化的指标包括脆弱性影响严重程度和脆弱性可利用性的9个影响因子，共10个测定级指标。下面以默认安装情况为例，说明测定级指标的量化方法。

软件默认安装情况可以分成默认安装、常常安装、有时安装和很少安装等4种情况，如表1所列。

表1 默认安装情况量化调查表

默认安装情况	默认安装	常常安装	有时安装	很少安装
评价语言值	x_1	x_2	x_3	x_4

用 $d_j(j=1,2,3,4)$ 分别表示这4个评估语言，用 x_j 表示对应评价语言 d_j 的评估语言值。设有 k 位专家参加调查，k 位专家中对软件的默认安装情况的评估隶属于评估语言等级 d_j 的数量为 $E_j(j=1,2,3,4)$，$\sum_{j=1}^{4}E_j=k$，则软件的默认安装情况的评估对评估语言等级 d_j 的隶属度 r_j 的计算方法如下：

$$r_j=\frac{E_j}{k} \tag{1}$$

最后，得到软件的默认安装情况的量化值 D：

$$D=\sum_{j=1}^{4}(r_j\times x_j) \tag{2}$$

3.3 基于模糊关系矩阵的指标权重的确定

在计算了各测定级指标的量化值之后，根据评估的递阶层次结构，计算各指标的权重，即确定各级测评因素的重要性。在脆弱性危险程度评估中，要确定的权重有3组，包括：1)脆弱性可利用性的影响因子在脆弱性可利用性中的评估权重值；2)脆弱性影响严重程度在脆弱性危险程度的评估权重值；3)脆弱性可利用性在脆弱性危险程度的评估权值。

以脆弱性可利用性的影响因子在脆弱性可利用性中的评估权重值为例，说明指标权重的确定方法。

脆弱性可利用性评估指标组成的指标集包含默认安装情况、稳定攻击代码情况等9个指标，表示为：

$$Z=\{z_1,z_2,\cdots,z_9\} \tag{3}$$

第一步 按照指标集 Z 中各元素对上层指标的重要性，

构建二元比较的定性排序标度矩阵 E：

$$E=\begin{bmatrix} e_{11} & e_{12} & \cdots & e_{19} \\ e_{21} & e_{22} & \cdots & e_{29} \\ \vdots & \vdots & \ddots & \vdots \\ e_{91} & e_{92} & \cdots & e_{99} \end{bmatrix} \tag{4}$$

式中，元素 $e_{kl}(k,l=1,2,\cdots,9)$ 可以取值 0，0.5 和 1，分别表示 z_k 比 z_l 重要，z_k 与 z_l 同等重要 z_k 没有 z_l 重要。

第二步　对矩阵 E 进行一致性检验，保证在二元比较过程中，判断思维不会出现矛盾。检验条件为：

$$\begin{cases} \text{if } e_{hk}>e_{hl}, & \text{then } e_{lk}>e_{kl} \\ \text{if } e_{hk}<e_{hl}, & \text{then } e_{lk}<e_{kl} \\ \text{if } e_{hk}=e_{hl}, & \text{then } e_{lk}=e_{kl} \end{cases} \tag{5}$$

检验条件 1 表示，对指标 z_h 来说，如果 z_h 比 z_k 的重要性大于 z_h 比 z_l 的重要性，则可以判断 z_k 一定没有 z_l 重要。

检验条件 3 表示，如果指标 z_h 与 z_k 同等重要，同时 z_h 与 z_l 也同等重要，则可以判断 z_k 与 z_l 也同等重要。

若不能满足上述的一致性检验条件，则需重新调整排序标度，重复步骤 1 和 2。

第三步　计算矩阵 E 的各行元素之和，按照大小排序即确定了指标重要性的定性排序关系。将定性排序的最重要指标与其他指标逐一进行二元对比，应用经验知识，逐一确定各指标的非归一化权重值。

第四步　对权重值归一化，得到归一化指标权向量：

$$\omega=(\omega_1,\omega_2,\cdots,\omega_9) \tag{6}$$

式中，评价指标数为 9，且满足：$\sum_{i=1}^{9}\omega_i=1$。

3.4　软件脆弱性危险程度的综合评价

通过聚合评价指标，获取最终的评价结果。在评估脆弱性危险程度的过程中，共需要两次聚合评估，分别评估脆弱性可利用性和脆弱性危险程度。

脆弱性可利用性的量化值为下层可利用性的各影响因子指标的加权总和，计算公式如下：

$$P=\sum_{s=1}^{9}(W_{ps}\times T_s) \tag{7}$$

式中，P 为脆弱性可利用性，s 为脆弱性可利用性的影响因子，W_{ps} 为各影响因子 s 的权重，T_s 为影响因子 s 的量化值。

软件脆弱性危险程度的量化值为脆弱性影响严重程度和脆弱性可利用性的加权总和，计算公式为：

$$C=W_s\times D+W_p\times P \tag{8}$$

式中，C 为软件脆弱性危险程度，D 为脆弱性影响严重程度，P 为脆弱性可利用性的量化值，W_s，W_p 分别为脆弱性影响严重程度和脆弱性可利用性的权重。

4　评估模型的实现

软件脆弱性危险程度评估模型在大规模特定域网主动防御系统中设计并实现。大规模特定域网是指由一个机构针对特定应用领域提供的网络，该网络具有集中的管理域，有边界，并且具有一定的服务器规模和用户规模。系统功能可以概括成态势感知、风险评估和主动防御 3 部分。系统结构如图 2 所示，图中仅强调与软件脆弱性危险程度评估相关的部分。

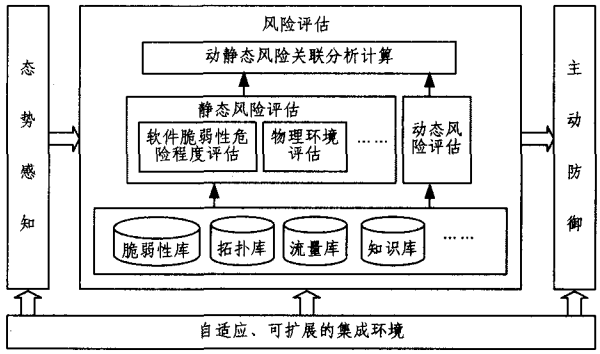

图 2　大规模特定域网主动防御系统结构

首先，综合采用防病毒网关解析、木马通信检测、DDoS 检测、入侵检测、主动蜜罐恶意代码捕获等多种手段，实现对网络安全态势的精确感知；然后进行风险评估。风险评估的基础是记录了软件脆弱性、网络拓扑结构、网络实时流量等信息的数据库。其中脆弱性数据库中要记录每一个脆弱性的完整信息。由于软件脆弱性非常复杂，相关的信息很多，许多信息本身很难准确定义和描述，相互之间的关系复杂，因此脆弱性数据库设计的关键是建立完善的脆弱性信息模型。目前的信息模型分成基本信息、关键属性信息和处理信息 3 部分，其中软件脆弱性影响和可利用性都属于关键属性信息，对应的脆弱性数据库管理界面截图如图 3 所示。

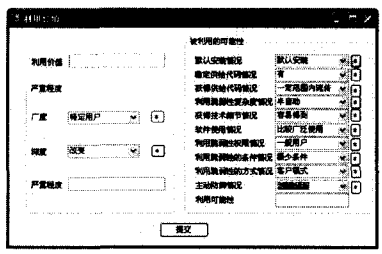

图 3　脆弱性数据库可利用性配置界面

风险评估在对各种数据进行融合和关联分析[13]的基础上，采用动静态结合的量化风险评估方法来实现。关键步骤如下：

1）建立脆弱性等关键的数据库。

2）依据本文的评估模型，基于脆弱性数据库中严重程度和可利用性等信息，对软件脆弱性危险程度进行评估；同时对物理环境、管理、人员等进行静态评估。

3）根据安全事件的攻击特征等，对网络安全进行动态风险评估。

4）综合考虑实时网络安全状态与静态评估结果，通过动静态风险的关联分析，建立各因素的关联关系，实现对当前网络安全风险的量化评估[14]。

结束语　软件脆弱性的危险程度是对软件脆弱性被利用来攻击系统的潜在危险的度量。对软件脆弱性危险程度进行合理的评估是脆弱性分析的关键技术之一。目前已知的一些定性分析方法，只有评价结果，没有评估过程。本文对脆弱性危险程度的量化评估进行了研究，提出了根据脆弱性影响的严重程度和脆弱性可利用性来对脆弱性危险程度进行评估的分析框架，并基于模糊理论提出软件脆弱性危险程度评估的量化模型，该模型在大规模特定域网主动防御系统的风险评

（下转第 216 页）

$X_F^0=\{(A,day,1)\}$

$Y_F^0=\{$[illegible]$\}$

$X_F^1=\{(A,day,1),(B,day,0.4),(D,day,0.8)\}$

$Y_F^1=\{(B,day,0.4),(D,day,0.7),$[illegible]$\}$

$X_F^2=\{(A,day,1),(B,day,0.4),(D,day,0.7),(C,day,0.4),(E,week,0.7)\}$

$Y_F^2=\{B,day,0.4),(D,day,0.5),(C,day,0.4),(E,week,0.5),$[illegible]$\}$

$X_F^3=\{(A,day,1),(B,day,0.4),(D,day,0.5),(C,day,0.4),(E,week,0.5),(G,week,0.5)\}$

$Y_F^3=\{(B,day,0.4),(D,day,0.5),$[illegible]$,(E,week,0.5),(G,week,0.5)\}$

$X_F^4=\{(A,day,1),(B,day,0.4),(D,day,0.5),(C,day,0.4),(E,week,0.5),(G,week,0.5)\}$

$X_F^4=X_F^3$

从上述的模糊/时态向量特性分析，可以得到模糊/时态数据库的的函数依赖特性，以便模糊/时态建模。

结束语 对于具有模糊时态函数依赖的数据库规范化理论来说，解决多粒度的模糊时态数据库的分解和优化问题，FTFD的关键是解决多粒度时态偏序性和模糊属性间依赖关系的判定。通过分析属性集的有限闭包、时态类型集的封闭集、属性集在给定时态上的依赖等概念，得到了模糊/时态向量的特征描述，对算法的可终止性、正确性进行了证明。后续的工作将从算法的多项式时间和效率上考虑能解决模糊时态依赖的有效判定和无损分解问题。

参 考 文 献

[1] 唐常杰. 时态数据库的沿革、特色与代表人物——时态数据库二十年回顾之一[J]. 计算机科学，1999，26(02)：27-30

[2] Clifford J. A Model for Historical Databases[C]// Workshop on Logical Bases for Database. Toulouse, France, 1982

[3] Jensen C S, Snodgrass R T, Soo M D. Extending existing dependency theory to temporal databases[J]. IEEE Trans. Knowledge and Data Engineering, 1996, 8(4): 563-582

[4] Wijsen J. Design of temporal relational databases based dynamic and temporal functional dependencies[C]// Proc. of the Int'l Workshop on Recent Advances in Temporal Databases. New York: Springer-Verlag, 1995: 61-76

[5] Wang X S, Bettini C, Jajodia S. Logical design for temporal databases with multiple granularities. ACM Trans[J]. Database System, 1997, 22(2): 115-170

[6] 郝忠孝. 时态数据库设计理论[M]. 北京：科学出版社，2009

[7] 郝忠孝，李艳娟. 时态函数依赖多值依赖混合集的成员籍问题研究[J]. 计算机研究与发展，2006，43(7)：1267-1272

[8] Fan Wen-fei, Siméon J. Integrity constraints for XML[C]// the nineteenth ACM SIGMOD SIGACT. 2000: 23-34

[9] Tian Jia-shen, Liu Ji-xue, Pan Wei-dong, et al. Performance Analysis and Improvement for Transformation Operators in XML Data Integration[C]// APWeb. 2008: 214-226

[10] Vincent M W, Liu Ji-xue, Mohania M K. On the equivalence between FDs in XML and FDs in relations[J]. Acta Inf, 2007, 44 (3/4): 207-247

[11] Vincent M W. Detecting privacy violations in database publishing using disjoint queries[C]// EDBT. 2009: 252-262

[12] 邓立国，马宗民. 模糊时态数据库关系代数演算规则分析[J]. 小型微型计算机系统，2009，30(12)：2433-2438

(上接第 172 页)

估中得到实现和应用。目前模型还在进一步完善中，包括脆弱性可利用性因素的扩展和细化、各指标及权重的完善等。

参 考 文 献

[1] Li Yi, Li Xin-ming. A New Taxonomy of Linux/Unix Operating System and Network Vulnerabilities[J]. Journal of Communication and Computer, 2006, 3(8): 16-19

[2] Piesses F. A Taxonomy of Causes of Software Vulnerabilities in Internet Software[C]// Proc. of the 13th International Symposium on Software Reliability Engineering. Annapolis, USA, 2002

[3] 王秋艳，张玉清. 一种通用漏洞评级方法[J]. 计算机工程，2008，34(19)：133-136

[4] Cima S. Vulnerability Assessment[R]. SANS Institute. 2001

[5] Microsoft Security Response Center Security Bulletin Severity Rating System[R]. Microsoft Security Response Center. 2002

[6] 黄明，曾庆凯. 软件脆弱性分类属性研究[J]. 计算机工程，2010，36(1)：184-186

[7] Du Wenliang, Aditya P. Categorization of Software Errors that led to Security Breaches[C]// Proc. of the 21st National Information System Security Conference. Virginia, USA, 1998

[8] Bishop M, Bailey D. Critical Analysis of Vulnerability Taxonomies[R]. CSE-96-11. Department of Computer Science at the University of California, 1996

[9] Krsul I. Software Vulnerability Analysis[D]. Purdue University, 1998

[10] Song Guang-feng, Mandujano S. CERIAS Classic Vulnerability Database User Manual[R]. CERIAS-TR-2000-17. Purdue University, 2000

[11] Bazaz A, Arthur J D. Towards A Taxonomy of Vulnerabilities [C]// Proceeding of the 40th Hawaii International Conference on System Sciences. Hawaii, USA, 2007

[12] 杜经农，卢炎生. 一种 Web 软件安全漏洞分类方法[J]. 计算机工程与应用，2009，45(25)：10-14

[13] 夏阳，陆余良. 计算机主机及网络脆弱性量化评估研究[J]. 计算机科学，2007，34(10)：74-79

[14] Miao Jia-jia, Wen Yan, Li Ai-ping. Large Scale Security Log Sources Integration: An Ensemble Method[C]// Proc. of the Second KDD Workshop on Mining multiple Information Sources. Las Vegas, USA, 2008

[15] Cheng Wen-cong, Su Xi-shan, Jia Yan. Network Dynamic Risk Assessment Based on the Threat Stream Analysis[C]// Proc. of the 9th International Conference on Web-Age Information Management. Zhangjiajie, China, 2008