doi:10.11959/j.issn.1000-436x.2015026

基于马尔可夫的检查点可信评估方法

田俊峰，张亚姣
（河北大学 数学与计算机学院，河北 保定 071002）

摘 要：为了发现软件的脆弱点，通过动态监测行为，对软件及其模块在一段时间内运行的可信状况进行研究，提出了基于马尔可夫的检查点可信评估模型。模型通过在软件行为轨迹中织入若干检查点来反映软件运行的行为表现，然后对检查点可信程度进行等级划分，通过马尔可夫模型及检查点权重反映检查点可信情况，最后综合每个检查点的可信情况得到软件整体的可信性。实验结果表明该模型能够有效反映软件中各部分可信情况，验证了模型的合理性和有效性。

关键词：软件可信评估；检查点；马尔可夫模型；混合高斯分布
中图分类号：TP393.08 **文献标识码**：A

Checkpoint trust evaluation method based on Markov

TIAN Jun-feng, ZHANG Ya-jiao
(College of Mathematics and Computer, Hebei University, Baoding 071002, China)

Abstract: According to the trust evaluation of software and its modules after running for a period of time a checkpoint trust evaluation model was presented based on Markov to find the vulnerability of software by monitoring software behavior dynamically. The model reflected the software running situation by some checkpoints worked in the software behavior trace, then divided some trust levels to reflect the checkpoint trusted situation by the Markov model and the weight of checkpoint, and finally synthesized the checkpoint trusted situation to determine the software trust value. Experimental results showed that the model could effectively reflect the trusted situation of the various parts of the software, and verify the rationality and availability of the model.

Key words: software trust evaluation; checkpoint; Markov model; Gaussian mixture model

1 引言

软件的可信性一直是人们广泛关注的问题。早期的计算机系统侧重于硬件系统的可靠性研究，可靠性是指系统在规定的条件下，在规定的时间内完成规定功能的能力，它表明系统中若存在故障，只要不影响正常功能的执行和完成，系统仍然是可靠的[1]。现如今系统可靠能力已不能满足人们的要求，人们希望系统完成其功能的同时要保证其安全性。武汉大学张焕国教授[2]指出：可信≈可靠＋安全，这充分体现了计算机发展的趋势。现如今的可信 PC 技术已较为成熟，可信计算机得到普遍应用，而可信软件的研究相对滞后。软件可信是指软件系统能够按照其设定的目标所预期的方式运行，软件的行为和用户的预期相一致[3]。软件可信主要表现在软件行为可信上，体现在程序从执行开始到结束的行为轨迹与预期行为轨迹相一致。由于软件本身的设计缺陷或软件外部的恶意攻击可能导致软件产生故障，从而偏离预期行为轨迹，最终给人们的工作和生活带来不良影响甚至造成巨大损失。因此，研究软件行为的可信性具有重要意义。

收稿日期：2013-09-04；**修回日期**：2013-11-12
基金项目：国家自然科学基金资助项目（61170254）；河北省自然科学基金资助项目（F2012201145）；河北省高等学校科学技术研究重点基金资助项目（H2012029）
Foundation Items: The National Natural Science Foundation of China(61170254); The Natural Science Foundation of Hebei Province(F2012201145); The Science and Technology Research Key Project in Colleges and Universities of Hebei Province(H2012029)

软件可信评估是软件可信研究中的热点，受到国内外学者的高度关注。针对软件可信评估，目前已有不少成果，但其相关理论和方法仍处于研究阶段。Wang 等人[4]综合考虑了软件在身份、能力和行为等方面的可信属性，提出了一个 Internet 环境下软件的可信概念模型及可信保障框架。杨善林等人[5]提出一种基于效用和证据理论的可信软件评估方法。蔡斯博等人[6]提出一种支持软件资源可信评估的框架。沈国华等人[7]提出了一种通用的软件可信评估框架，适用于不同形态的软件。丁帅等人[8]提出一种需求驱动的考虑可信属性间关联的可信性评估及演化模型。上述成果主要以静态的方式从软件的可信属性、证据理论和评估需求等方面对软件进行评估，对软件运行时的动态信息关注较少，没有考虑软件的动态可信性。

当前，基于软件行为的可信研究也有不少成果。Diakov 等人[9]提出了一种基于 CORBA 中间件平台的软件行为监测框架。Li [10]提出了基于全局因果跟踪技术捕获多维软件系统行为的软件行为监测框架。Mariani 等人[11]构建了一个自动捕获构件行为的监测框架，使用构件包装器截取构件间的交互行为信息。Cheng 等人[12]提出了一种行为监测框架 MOP，根据给定的行为规约自动生成检测器动态监测软件系统。古亮等人[13]提出了基于 TPM 的运行时软件可信证据收集机制。刘玉玲等人[14]通过在软件行为轨迹中织入若干检查点，提出了一种基于软件行为的检查点风险评估信任模型。这些模型大多数是通过动态监测软件行为，然后依据其是否满足给定规约来判定软件一次运行的可信性，而通过动态监测行为，寻找软件的脆弱点，对软件及其模块在一段时间内运行的可信状况研究的很少。

针对上述问题，本文提出一种基于马尔可夫状态的软件可信评估方法，在软件行为轨迹中织入若干检查点，通过采集检查点处的场景信息来动态地反映软件运行中的行为表现，并对检查点进行可信等级划分以表征检查点的可信状态，通过马尔可夫模型分析检查点的可信情况，最终判断软件整体的可信程度。这种方法通用性较强，可以适用于不同形态的软件系统，并且可以有针对性地评估一段时间内软件及其模块的可信性，找出软件的脆弱点，便于加强其安全性以提高软件整体的可信性。

2　基于马尔可夫的软件可信评估模型

2.1　相关定义

定义 1　软件行为。指软件运行时作为主体，依靠其自身的功能对客体的施用、操作或者动作。任何软件的执行过程，都可看作是该软件的一系列软件行为，每个软件行为包括行为轨迹和检查点场景的集合。

定义 2　行为轨迹。将软件主体所实施的行为，按照时间顺序记录下来形成形式化序列，称为该软件主体的行为轨迹。

定义 3　检查点。软件流程上一些重要点，一般选取软件分支处和一个独立基本功能结束处设置，用以提取软件的场景信息。一个独立基本功能小则一个系统调用，大则一个功能模块调用。检查点设置的粒度越细，则软件检查的粒度越细，软件可信性判断越准确，但对软件运行效率的影响也就越大，因此实际应用中根据对准确性和运行效率的要求设置合适的检查点。所有检查点的集合用 E 表示。

定义 4　场景。检查点处必要的软件运行背景信息和结果信息，包括系统调用、系统调用参数、计算结果、CPU 使用率、内存占有率、软件执行的时间等，它是一个 n 元组，记为 S，$S=\{s_1,s_2,\cdots,s_n\}$。

定义 5　场景偏离值。表示检查点 e 的场景信息偏离正常范围的程度，记为 $P(e)$。

定义 6　信任度。在一段时间内，根据检查点处场景信息得到的软件行为可信程度的期望值。

2.2　可信评估模型

根据软件的可信性定义，如果一个软件系统的行为总是与预期相一致，那么该软件是可信的。本文提出的基于马尔可夫的可信评估模型（MTEM, Markov trust evaluation model）是在软件行为轨迹中织入若干检查点，根据检查点的场景信息判断该检查点的状态。当一个软件开发完成后，经过多次正常的训练就能得到该软件正常运行时场景中各属性的取值范围；当软件实际运行时，检查点处的场景值会发生偏差，偏差越大，软件出现异常的可能性就越大，当偏差超过某个阈值时说明软件发生异常。

如图 1 所示，本文提出的软件可信评估方法是综合了检查点可信分析和检查点权重 2 方面内容判断软件的整体可信性。一方面，通过收集软件运行过程中每一个检查点的状态信息，分析每一个检查

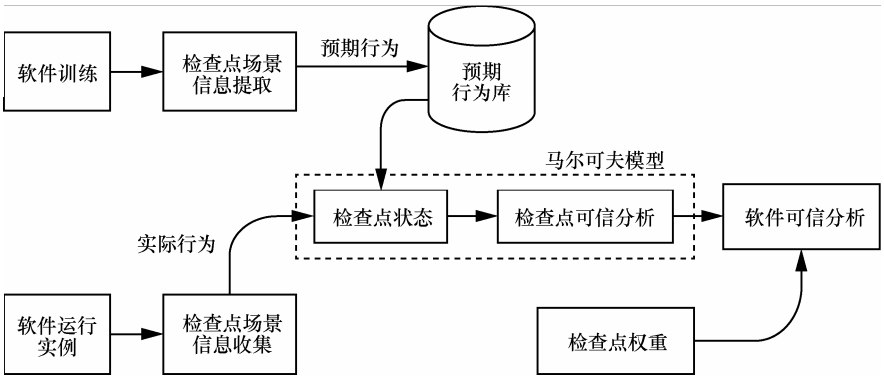

图 1 基于马尔可夫的可信评估模型（MTEM）框架

点的可信情况；另一方面，分析软件运行过程中，所有检查点的状态情况对软件整体可信性的影响。

2.2.1 场景信息表示

本文假设场景信息服从混合高斯分布。混合高斯模型[15]是聚类算法和概率密度拟合中的常用模型，其基本假设是待研究数据服从混合高斯分布，或者说数据可以由数个高斯分布生成，多数自然数据具有分类聚集性质，由中心极限定理可知，这样的假设是合理的。

混合高斯模型本身可以变得任意复杂，通过增加高斯模型的个数可以任意地逼近任何连续的概率分布，具体地说，每个混合高斯模型由 M 个高斯分布组成。第 i 个高斯模型记为 $X_i \sim N_n(\mu_i,\Sigma_i)$，根据高斯分布的可加性，有 $\sum_{i=1}^{M} k_i X_i \sim N_n\left(\sum_{i=1}^{M} k_i\mu_i, \sum_{i=1}^{M} k_i^2\Sigma_i\right)$，$k_i$ 是第 i 个高斯模型对应的权值。那么对于本模型来说，检查点 e 的场景信息可表示为 $S(e)\sim N_n(\mu(e),\Sigma(e))$，其中，$\mu(e)$，$\Sigma(e)$ 分别是检查点 e 处的 $\sum_{i=1}^{M} k_i\mu_i, \sum_{i=1}^{M} k_i^2\Sigma_i$，通常采用 EM 算法[16]求解混合高斯分布的参数值，即 k_i,μ_i,Σ_i 的值。k_i 表示训练样本与第 i 个高斯模型匹配的概率；μ_i 表示第 i 个高斯模型的均值；Σ_i 表示第 i 个高斯模型的方差，反映样本的偏差程度。

检查点 e 处的场景偏离值由该检查点处的场景信息与其均值的欧氏距离(Euclidean distance)表示，该均值即为上述的混合高斯模型中的参数值 μ_i。

$$P(e)=\sqrt{\sum_{i=1}^{n}[X(e)_{s_i}-\mu(e)_{s_i}]^2} \tag{1}$$

由式(1)可知，$P(e)\geqslant 0$。理论上讲，当 $P(e)=0$ 时，场景信息的取值是最佳情况，$P(e)$ 越大，表示检查点 e 越不可信。通过正常的训练可以得到检查点 e 的场景偏离值 $P(e)$ 的取值情况：$P(e)\leqslant \max P(e)$。当 $P(e)\leqslant \max P(e)$，说明检查点 e 的场景信息是正常的，当 $P(e)>\max P(e)$ 时，说明检查点 e 存在风险。将风险分为 3 种程度。

1) 弱风险。检查点的风险非常小，在进行可信度量时可以忽略不计。

2) 中度风险。检查点存在较大风险，此时的检查点状态对软件可信性会产生一定的影响。

3) 强风险。检查点已经出现异常，此时软件整体也不可信。

为了区分 3 种风险程度，设置弱风险因子 α 和强风险因子 β 这 2 个指标。当 $P(e)\leqslant\alpha$ 时，场景信息属于弱风险程度；当 $P(e)\in(\alpha,\beta)$ 时，场景信息属于中度风险程度；当 $P(e)\geqslant\beta$ 时，场景信息属于强风险程度。这 2 个指标可以根据训练情况确定。

2.2.2 检查点的马尔可夫模型

马尔可夫模型[17]是一种统计模型，它是运用马尔可夫链的理论和方法来研究分析有关马氏过程数据的变化规律。本文采用马尔可夫模型分析检查点的可信情况，这是因为马尔可夫模型具有预测能力，它能够根据历史数据发现检查点可信情况变化的规律，推测未来检查点可信情况变化的趋势。马尔可夫链的基本概念是系统的状态和状态的转移，该过程中，系统由当前状态转移至下一个状态时存在转移概率，并且这种概率只与系统当前状态有关，而与过去的状态无关，这一特性就是马尔可夫的无后效性。

马尔可夫模型处理问题的一般步骤为：首先根据系统各部分状态变化的规律，给出系统马尔可夫状态转移图；然后根据系统状态转移图列出系统状态转移概率矩阵；最后根据状态方程求解各状态概率。以下将遵守上述步骤建立检查点的马尔可夫模

型来评估其可信性。

人们通常采用自然语言描述信任等级，借鉴文献[18]的思想，本文将检查点可信情况划分为 4 个等级：绝对可信、一般可信、临界可信和不可信，用 k_0、k_1、k_2、k_3 表示，这 4 个等级也就是检查点的 4 种状态。下面以检查点 e 为例。

1) 当 $P(e)\in(0,\max P(e)]$ 时，检查点处于绝对可信状态，表明检查点的场景信息完全在正常范围内，软件实际运行时所表现的行为与预期行为完全一致。

2) 当 $P(e)\in(\max P(e),\alpha]$ 时，检查点处于一般可信状态，表明检查点的场景信息存在弱风险，软件实际运行时所表现的行为与预期行为基本一致，可能存在极小甚至可以忽略的差别。

3) 当 $P(e)\in(\alpha,\beta)$ 时，检查点处于临界可信状态，表明检查点的场景信息存在中度风险，软件实际运行时所表现的行为与预期行为接近但不完全一致，软件可能是可信的，也可能是不可信的，这时就需要进行进一步的分析。

4) 当 $P(e)\in[\beta,\infty)$ 时，检查点处于不可信状态，表明检查点的场景信息存在强风险，软件实际运行时所表现的行为与预期行为完全不一致，软件是不可信的。

软件运行过程中，若某一检查点出现不可信状态，立即停止运行，记录足够多次运行过程中各检查点的状态信息，建立马尔可夫模型，如图 2 所示。

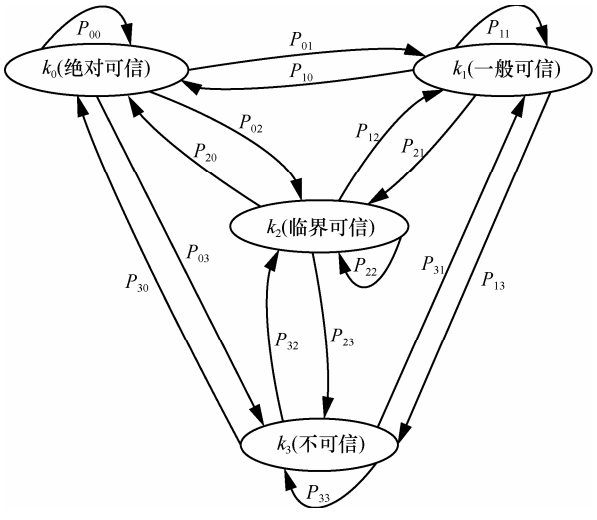

图 2　马尔可夫状态转移

分析所得数据，求出各状态的转移概率，建立马尔可夫状态概率转移矩阵 $\boldsymbol{P}_e$

$$\boldsymbol{P}_e=\begin{bmatrix}P_{00} & P_{01} & P_{02} & P_{03}\\ P_{10} & P_{11} & P_{12} & P_{13}\\ P_{20} & P_{21} & P_{22} & P_{23}\\ P_{30} & P_{31} & P_{32} & P_{33}\end{bmatrix}$$

根据马尔可夫链遍历性特点，系统经过长时间运行后会存在一个稳定的状态，设检查点 e 的稳态概率向量为 $\boldsymbol{y}_e=(y_0,y_1,y_2,y_3)$，求解 $\boldsymbol{y}_e$ 值如式(2)所示，那么稳态概率向量 $\boldsymbol{y}_e$ 的各元素值就是检查点 e 处于稳态时各状态的概率值。

$$\begin{cases}\boldsymbol{y}_e p_e=\boldsymbol{y}_e\\ \sum_{i=0}^{3}y_i=1\end{cases}\tag{2}$$

2.2.3　检查点权重计算

当某一检查点出现不可信状态时，说明软件出现异常，立即停止运行；当任意检查点都没有出现不可信状态时，软件运行结束会出现正常和异常 2 种情况，此时软件异常可能是因为某些检查点处于临界可信状态，当临界可信状态的检查点累积到一定数量或程度表明软件出现异常，这时就需要分析出现临界可信状态的检查点对整个软件的影响。设临界可信状态影响向量为 $\boldsymbol{w}$，其中的元素 w_e 表示检查点 e 处于临界可信状态对软件的影响程度。

检查点出现绝对可信或一般可信状态时都视为检查点是可信的，这些检查点的累积不会对软件可信有影响；检查点出现不可信状态时认为软件出现异常，这种检查点对软件可信的影响是绝对的，即只要有一个检查点出现不可信状态，软件就异常；检查点出现临界可信状态时说明软件存在风险，这时检查点 e 对软件可信的影响程度为 w_e。

向量 $\boldsymbol{N}_e=\{N_e^{00},N_e^{01},N_e^{10},N_e^{11}\}$ 记录检查点 e 的信息，其中，N_e^{00} 表示软件异常时，检查点 e 临界的次数；N_e^{01} 表示软件异常时，检查点 e 可信的次数；N_e^{10} 表示软件正常时，检查点 e 临界的次数；N_e^{11} 表示软件正常时，检查点 e 可信的次数。那么，软件异常时，检查点 e 处于临界状态的概率为 $\eta_e^0=\dfrac{N_e^{00}}{N_e^{00}+N_e^{01}}$，表示检查点 e 临界对软件异常的影响；软件正常时，检查点处于临界状态的概率为 $\eta_e^1=\dfrac{N_e^{10}}{N_e^{10}+N_e^{11}}$，表示检查点 e 临界对软件正常的影响。当 η_e^0 与 η_e^1 相近时，说明检查点 e 处于临界状态

时对软件的影响程度较小；当$\eta_e^0 > \eta_e^1$且差距越大，说明检查点 e 临界时软件异常的可能性越大；当$\eta_e^1 > \eta_e^0$且差距越大，说明检查点 e 临界时软件异常的可能性越小，因此，用$\eta_e = \eta_e^0 - \eta_e^1$，$\eta_e \in [-1,1]$反映检查点 e 临界的影响程度。

本文采用模糊层次分析法[19]求解权重值，它是一种结合了层次分析法和模糊集合理论的决策方法，其原理是通过描述任意 2 个因素之间关于某种准则的相对重要程度，构造模糊一致矩阵，用于因素权重的确定。本文中求解权重，首先要对检查点的临界影响程度大小进行两两比较，然后利用极差变换的思想对数据进行标准化处理，形成模糊一致矩阵，其中

$$r_{ij} = \frac{1}{4}\left(\eta_{e_i} - \eta_{e_j}\right) + 0.5 \tag{3}$$

再依据式(4)求解权重。

$$\lambda_i = \frac{1}{n} - \frac{1}{2a} + \frac{1}{na}\sum_{j=1}^{n} r_{ij},\ a \geqslant \frac{n-1}{2},\ i = 1,2,\cdots,n \tag{4}$$

2.2.4 软件可信度的计算

软件可信度的判断是在一段时间内，根据检查点处场景信息得到软件行为的可信程度，判断软件的整体可信性时需要综合检查点可信程度和检查点重要程度 2 方面内容，分别从横向和纵向 2 个角度考虑这 2 个问题。

1) 从横向上分析检查点的可信程度问题，收集软件运行过程中每一个检查点的状态信息，在检查点处运用马尔可夫模型，计算得到该检查点的各状态概率 $y_0(e)$、$y_1(e)$、$y_2(e)$、$y_3(e)$，表征了每一个检查点的可信情况，其中绝对可信状态概率 $y_0(e)$ 和一般可信状态概率 $y_1(e)$ 越大，临界可信状态概率 $y_2(e)$ 和不可信状态概率 $y_3(e)$ 越小，那么检查点 e 的可信度越高。

2) 从纵向上分析检查点的重要程度问题，分析软件一次运行过程中，所有检查点的状态情况对软件整体可信性的影响。本文中指出检查点处于绝对可信状态和一般可信状态时，对软件整体可信性没有影响；检查点处于不可信状态时，对软件整体可信性的影响是绝对的，即为 1；检查点处于临界可信状态时，对软件整体可信性的影响即为所求得的临界状态影响向量 $\boldsymbol{w}$，其中检查点 e 的临界状态影响权重 w_e 越大，说明该检查点处于临界状态时对软件整体的可信性影响越大，那么该检查点的可信度越低。

由上可得，检查点 e 的可信度为

$$T(e) = 1 - [\omega_e y_2(e) + y_3(e)] \tag{5}$$

根据木桶理论[20]，用一个木桶来装水，如果组成木桶的木板参差不齐，那么它能盛水的容量不是由这个木桶中最长的木板来决定，也不是由所有木板的平均长度来决定，而是由这个木桶中最短的木板来决定，这又被成为“木桶效应”。因此，软件的可信度记为 $T = \min T(e)$，其中 $T(e)$ 最小的检查点 e 就是那块“短板”，即软件的脆弱点。

3 实验及分析

为了验证模型的合理性和有效性，在 CPU 为 Intel(R)Pentium(R)4 3.06 GHz，内存为 2 GB，Linux 内核版本为 2.4.20 的主机上进行实验。以 Gzip 解压文档文件为目标程序，设置 6 个检查点 $e_1, e_2, \cdots, e_6$。Gzip 中存在多个安全漏洞[21]，当用户解压了包含有特制解码表的文档，就可能会导致拒绝服务或执行任意代码。本文考虑 CVE-2006-4335 和 CVE-2006-4336 这 2 个漏洞。其中，CVE-2006-4335 漏洞是 unlzh.c 文件的 make_table()函数中存在的漏洞，将检查点 e_4 设置在 make_table()函数处，CVE-2006-4336 漏洞是 unpack.c 文件的 build_tree()函数中存在的漏洞，将检查点 e_5 设置在 build_tree()函数处。为了讨论方便，实验中选择比较直观的 CPU 使用率，内存占有率和磁盘读写次数作为场景信息。下面在以下 2 种情况下进行分析。

1) 一般情况

在 Gzip 多次正常运行时，捕获各检查点处的场景信息，在构建各检查点的马尔可夫模型以及分析检查点的权重基础上，计算检查点的可信度，结果如图 3 所示。可以看出，本文提出的 MTEM 模型检测到的各检查点可信度最小值为 0.58（检查点 e_4 的可信度），根据木桶理论，得到 Gzip 软件的可信度为 0.58。

将本模型与另一个基于检查点的评估模型 CBRA-TM 模型[14]进行比较。为确保运行背景一致，两模型对软件多次正常运行的检查点处的同一组场景数据进行分析。图 3 中显示了 CBRA-TM 模型采用大参数和小参数 2 种情况下 Gzip 软件多次运行时各检查点的平均可信度（即对任一检查点，多次运行得到检查点可信度的平均值），由于参数设

置的不同，所得检查点的可信度也有所不同。MTEM 模型所得结果反映了一段时间内各检查点的可信状况，与 CBRA-TM 模型反映的检查点的可信状况基本一致，且每个检查点的可信度大小比较结果是相同的，说明本模型能够体现出软件各检查点的可信状况，评估结果是合理的。CBRA-TM 模型因受参数设置的影响，计算所得的检查点可信度会受到影响。MTEM 模型的所得数据都来源于实验，受人为设置参数或规约的影响较小。

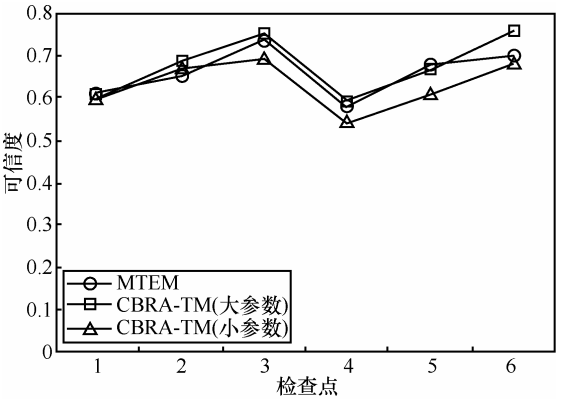

图 3　MTEM 模型和 CBRA-TM 模型软件可信度变化

2) 异常情况

当 Gzip 软件针对 CVE-2006-4335 或 CVE-2006-4336 漏洞遭受攻击时，多次运行捕获各检查点处的场景信息，得到的各检查点的可信度如图 4 所示。当 Gzip 软件针对 CVE-2006-4335 漏洞遭受攻击时，检查点 e_4 的可信度降低为 0.40，此时 Gzip 软件的可信度为 0.40；当针对 CVE-2006-4336 漏洞遭受攻击时，检查点 e_5 的可信度降低为 0.35，此时 Gzip 软件的可信度为 0.35。这说明由于环境的变化特别是当软件遭受攻击时，场景信息会发生变化，而这一变化可以通过马尔可夫模型体现出来，对应检查点及软件的可信度变得很低，因此 MTEM 模型能够准确地反映软件的脆弱点。同时，该实验也可以体现出模型具有较强的通用性和针对性，只要把检查点设置在某个模块中就能较为直观地反映该模块的可信状况，从而针对软件的脆弱点来提高软件整体的可信性。

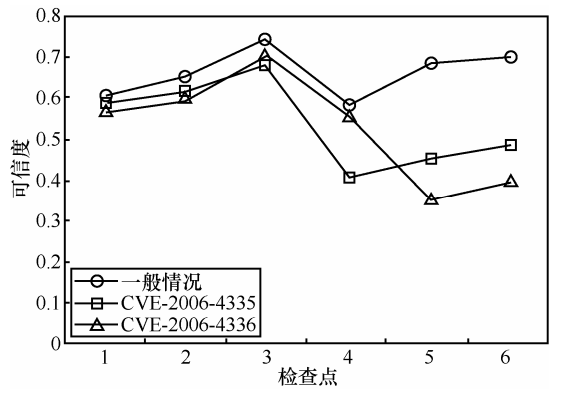

图 4　MTEM 模型不同环境下软件可信度变化

4　结束语

本文提出了基于马尔可夫的软件可信评估模型，它是一种基于软件行为的可信评估方法，通过将检查点织入软件行为轨迹中收集软件运行时的场景信息，动态地反映软件的运行情况，进一步将检查点情况划分为 4 个可信等级，表征检查点的可信状态并通过马尔可夫模型分析检查点的可信情况，最后找出软件的脆弱点，便于加强其安全性以提高软件整体的可信性。实验表明该模型能够较准确和合理的计算软件的可信度。在以后的工作中，将要对软件中各模块的价值进行分析，进一步提高软件可信度研究的合理性和精确度。

参考文献：

[1] 尚珊珊, 赵轶群. 软件可靠性综述[J]. 软件导刊, 2006,5(15):3-5.
SHANG S S, ZHAO Y Q. Overview of software reliability[J]. Software Guide, 2006,5(15):3-5.

[2] ZHANG H G, LUO J, JIN G, *et a1*. Development of trusted computing research[J]. Journal of Wuhan University (Natural Science Edition), 2006, 52(5):513-518.

[3] Trusted Computing Group. TCG specification architecture overview[EB/OL].https://www.trustedcomputinggroup.org/groups/TCG_1_0_Architecture_overview.pdf.2013.

[4] WANG H M, TANG Y B, YIN G, *et al*. Trustworthiness of Internet-based software[J]. Science in China: Series F Information Sciences, 2006, 49(6):759-773.

[5] 杨善林, 丁帅, 褚伟. 一种基于效用和证据理论的可信软件评估方法[J]. 计算机研究与发展, 2009, 46(7): 1152-1159.
YANG S L, DING S, CHU W. Trustworthy software evaluation using utility based evidence theory[J]. Journal of Computer Research and Development, 2009, 46(7):1152-1159.

[6] 蔡斯博, 邹艳珍, 邵凌霜等. 一种支持软件资源可信评估的框架[J]. 软件学报, 2010, 21(2): 359-372.
CAI S B, ZOU Y Z, SHAO L S, *et al*. Framework supporting software assets evaluation on trustworthiness[J]. Journal of Software, 2010, 21(2): 359-372.

[7] 沈国华, 黄志球, 钱巨等. 软件可信评估模型及其工具实现[J]. 计算机科学与探索, 2011, 5(6): 553-561.
SHEN G H, HUANG Z Q, QIAN J, *et al*. Research on software trustworthiness evaluation model and its implementation[J]. Journal of Frontiers of Computer Science an Technology, 2011, 5(6):553-561.

[8] 丁帅, 鲁付俊, 杨善林等. 一种需求驱动的软件可信性评估及演化

模型[J]. 计算机研究与发展, 2011, 48(4): 647-655.
DING S, LU F J, YANG S L, *et al*. A requirement-driven software trustworthiness evaluation and evolution model[J]. Journal of Computer Research and Development, 2011,48(4): 647-655.

[9] DIAKOV K, BATTERAMA J, ZANDBELT H. Monitoring of distributed component interactions[A]. Proceeding of IFIP International Conference on Distributed Systems Platforms and Open Distributed Processing[C]. New York, USA, 2000.229-243.

[10] LI J. Monitoring and characterization of component-based systems with global causality capture[A]. Proceeding of the 23rd International Conference on Distributed Computing Systems[C]. Rhode Island, USA, 2003.422-433.

[11] MARIANI L, PEZZE M. A technique for verifying component-based software[J]. Electronic Notes in Theoretical Computer Science, 2005,116(1): 17-30.

[12] CHEN F, GRIGORE R. Mop: an efficient and generic runtime verification framework[A]. Proceeding of the 22nd Annual ACM SIGPLAN Conference on Object-Oriented Programming Systems and Applications[C]. Montreal, Canada, 2007.569-588.

[13] 古亮, 郭耀, 王华等. 基于 TPM 的运行时软件可信证据收集机制[J]. 软件学报, 2010, 21(2): 373-388.
GU L, GUO Y, WANG H, *et al*. Runtime software trustworthiness evidence collection mechanism based on TPM[J]. Journal of Software,2010, 21(2): 373-388.

[14] 刘玉玲, 杜瑞忠, 冯建磊等. 基于软件行为的检查点风险评估信任模型[J]. 西安电子科技大学学报, 2012, 39(1): 179-184.
LIU Y L, DU R Z, FENG J L, *et al*. Trust model of software behaviors based on check point risk assessment[J]. Journal of Xidian University, 2012,39(1):179-184.

[15] 刘辉, 白峰杉. 基于混合高斯过程模型的高光谱图像分类算法[J]. 高校应用数学学报, 2010, 25(4): 379-385.
LIU H, BAI F S. Hyperspectral image classification based on mixed Gaussian process model[J]. Applied Mathematics A Journal of Chinese Universities, 2010,25(4):379-385.

[16] 王爱萍, 张功营, 刘方. EM 算法研究与应用[J]. 计算机技术与发展, 2009, 19(9): 108-110.
WANG A P, ZHANG G Y, LIU F. Research and application of EM algorithm[J]. Computer Technology and Development, 2009,19(9): 108- 110.

[17] 胡宇驰. 应用马尔科夫状态图法进行可靠性评估[J]. 电子科技大学学报, 2001, 30(2): 175-180.
HU Y C. Evaluation of reliability of a fault-tolerance computer system by Markov status graph[J]. Journal of UEST of China, 2001, 30(2):175-180.

[18] JØSANG A. A logic for uncertain probabilities[J]. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 2001, 9(3):279-311.

[19] 张吉军. 模糊层次分析法(FAHP)[J]. 模糊系统与数学, 2000, 12(2):80-88.
ZHANG J J. Fuzzy analytical hierarchy process[J]. Fuzzy Systems and Mathematics, 2000,12(2):80-88.

[20] 邵建平, 王玲. “拉长板”木桶原理及其运用研究[J]. 科技管理研究, 2005,4: 159-161.
SHAO J P, WANG L. Study on “stretched panels” bucket theory and its applicaion[J]. Science and Technology Management Research, 2005,4: 159-161.

[21] 国家信息安全漏洞平台. GNU GZip 文档处理多个安全漏洞[EB/OL]. http://www.cnvd.org.cn/sites/main/preview/ldgg_preview.htm?tid=CNVD-2009-09295,2013.
China National Vulnerability Database. GNU Gzip Deals with many security vulnerabilities[EB/OL]. http://www.cnvd.org.cn/sites/main/preview/ldgg_preview.htm?tid=CNVD-2009-09295,2013.

作者简介：

田俊峰（1965-），男，河北保定人，河北大学教授、博士生导师，主要研究方向为信息安全与可信计算。

张亚姣（1988-），女，河北秦皇岛人，河北大学硕士生，主要研究方向为信息安全与可信计算。