An Ideal of Hilbert Algebras in BCK-Algebras

ZHANG Qiuna, LI Dongmei, CHEN Weili, YANG Aimin, FENG Lichao

College of Light Industry & College of Science, Hebei Polytechnic University, Tangshan, China

e-mail: zhangqiuna@yahoo.com.cn

Abstract: In [3], it indicated that every Hilbert Algebras is anti-positive implicative BCK-algebras. Therefore, in this article, we give an ideal of Hilbert Algebras in BCK-algebras, as well as some propositions.

Keywords: ideal; Hilbert Algebras; BCK-algebras; anti-positive implicative

The notion of BCK-algebras was formulated first in 1966 by K. Iseki, Japanese Mathematician. This notion is originated from two different ways. One of the motivations is based on set theory; another motivation is from classical and non-classical propositional calculi. The notion of ideals in BCK-algebras was introduced by K. Iseki in 1975. The ideal theory plays a fundamental role in the general development of BCK-algebras, so in this article, we will give an ideal of Hilbert Algebras in BCK-algebras, and some propositions.

1. Introduction

Definition 1.1 [3]: Suppose H is a nonempty set, $\to$ is a binary operation on H, $1 \in H$. Then $(H, \to, 1)$ is Hilbert algebras if it satisfies the following conditions for any x, y, z in H:

H_1 $x \to (y \to x) = 1$;

H_2 $(x \to (y \to z)) \to$

$((x \to y) \to (x \to z)) = 1$;

H_3 If $x \to y = 1$ and $y \to x = 1$, then $x = y$.

Lemma 1.2 [3]: Suppose $(H, \to, 1)$ is a Hilbert algebras, the following conditions are satisfies for any x, y, z in H:

H_4 $x \to (y \to z) =$

$(x \to y) \to (x \to z)$;

H_5 $x \to (y \to z) = y \to (x \to z)$;

H_6 $(x \to y) \to$

$((y \to z) \to (x \to z)) = 1$;

H_7 $x \to ((x \to y) \to y) = 1$;

H_8 $x \to x = 1$;

H_9 $1 \to x = 1$.

We can find the proof in [2] and [3].

Given a Hilbert algebras $(H, \to, 1)$, we can define a binary operations*and two binary relations $\prec, \prec'$ on H.

$$x * y = y \to x, x \prec y \Leftrightarrow x \to y = 1, x \prec' y \Leftrightarrow x * y = 1,$$

then, we know $x \prec' y$ in $(H, *, 1)$ if and only if $y \prec x$ in $(H, \to, 1)$. *and $\to$ are two opposite binary operations on H, $\prec$ and $\prec'$ are two opposite order relations on H.

Lemma 1.3 [3]: $(H, \to, 1)$ is Hilbert algebras if and only if $(H, *, 1)$ is a positive implicative BCK-algebra.

We can find the proof in [3]. 1 is a minimal element under the order of BCK-algebras in H. So we can change the definition of ideal in [1] into Definition 2.1.

2. Ideal and Propositions

Definition 2.1 Let $(H, \to, 1)$ be BCK-algebras and let I be a nonempty subset of H. Then, I is called an ideal of H, if for all x, y in H.

1) $1 \in I$,

2) $y \to x \in I$ and $y \in I$ imply $x \in I$.

Definition 2.2 Given Hilbert algebras $(H, \to, 1)$, a nonempty subset I of H is said to be a Hilbert ideal if it satisfies for all x, y, z in H.

1) $1 \in I$,

2) $z \to (y \to x) \in I$ and $z \to y \in I$

imply $z \to x \in I$.

Theorem 2.1 Any Hilbert ideal must be an ideal, but the inverse is not true.

Proof: Suppose I is a Hilbert ideal. If $y \to x \in I$ and $y \in I$, then

$1 \to (y \to x) = y \to x \in I$, and $1 \to y = y \in I$ by Definition 2.1

$1 \to x = x \in I$, thus I is an ideal. The inverse is not true, for example let $H = \{1,2,3\}$ in which $\to$ is given by the table:

Then $(H, \to, 1)$ is a BCK-algebra. $\{1,2\}$ is an ideal of H, but not a Hilbert ideal.

Table 1

$\rightarrow$	1	2	3
1	1	1	1
2	2	1	2
3	3	3	1

Theorem 2.2 Suppose I is a nonempty subset of a BCK-algebras H, then the following conditions are equivalent:

(a) I is a Hilbert ideal;

(b) I is an ideal, and for any x, y in H, $y \to (y \to x) \in I$ implies $y \to x \in I$;

(c) I is an ideal, and for any x, y, z in H, $z \to (y \to x) \in I$ implies $(z \to y) \to (z \to x) \in I$;

(d) $1 \in I$ and $z \to (y \to (y \to x)) \in I, z \in I$ imply $y \to x \in I$.

Proof: $(a) \Rightarrow (b)$ If I is a Hilbert ideal, by Theorem 2.3, I is an ideal. Suppose $y \to (y \to x) \in I$, since $y \to y = 1 \in I$, by Definition 2.2, $y \to x \in I$, (b) holds.

$(b) \Rightarrow (c)$ Assume (b) and $z \to (y \to x) \in I$

$z \to (z \to ((z \to y) \to x))$

$(b) \Rightarrow (c) = z \to (z \to (y \to x))$

$= (z \to z) \to (z \to (y \to x)) =$

$1 \to (z \to (y \to x)) = z \to (y \to x) \in I$, it follows that $z \to (z \to ((z \to y) \to x)) \in I$, by (b) $z \to ((z \to y) \to x) \in I$. As $(z \to y) \to (z \to x) = z \to ((z \to y) \to x)$, then $(z \to y) \to (z \to x) \in I$. which is (c).

$(c) \Rightarrow (d)$ It's clear that $1 \in I$.

If $z \to (y \to (y \to x)) \in I, z \in I$, then

$y \to (y \to (z \to x))$

$= y \to (z \to (y \to x))$

$z \to (y \to x) = y \to (z \to x)$

$= 1 \to (y \to (z \to x))$

$= (y \to y) \to (y \to (z \to x))$

$= y \to (y \to (z \to x)) \in I$,

since I is an ideal,

and $z \in I$ thus $y \to x \in I$. (d) holds.

$(d) \Rightarrow (a)$ First proof I is an ideal. Suppose $y \to x \in I$ and $y \in I$, then

$y \to (1 \to (1 \to x)) \in I$, and $y \in I$,

by (d) $1 \to x = x \in I$. i.e., I is an ideal. Next let

$z \to (y \to x) \in I$ and $z \to y \in I$,

$(z \to y) \to (z \to (z \to x)) = z \to (y \to (z \to x))$

$= y \to (z \to (z \to x))$

$= y \to ((z \to z) \to (z \to x))$

$= y \to (1 \to (z \to x)) = y \to (z \to x)$

$= z \to (y \to x) \in I$,

then $(z \to y) \to (z \to (z \to x)) \in I$.

Combining $z \to y \in I$ and using (d) $z \to x \in I$. This have proofed that I is a Hilbert ideal. Thus the proof is completed.

Theorem 2.3 Suppose A and B are ideals of Hilbert algebras in BCK-algebras H, and $A \subset B$, if A is a Hilbert ideal, so is B.

Proof: Let

$z \to (y \to x) \in B$

and denote $u = z \to (y \to x)$, then

$z \to (y \to (u \to x))$.

$= z \to (u \to (y \to x))$

$= u \to (z \to (y \to x)) = 1 \in A$

A is a Hilbert ideal, by making use of **Theorem 2.4** (c) we have

$(z \to y) \to (z \to (u \to x)) \in A$.

$(z \to y) \to (z \to (u \to x))$

$= (z \to y) \to (u \to (z \to x))$

$= u \to ((z \to y) \to (z \to x))$

$= (z \to (y \to x)) \to ((z \to y) \to (z \to x)) \in A$

$A \subset B$, then $(z \to (y \to x)) \to ((z \to y) \to (z \to x)) \in B$ is an ideal and $z \to (y \to x) \in B$, then it follows $(z \to y) \to (z \to x) \in B$.

This means that for the ideal B, $z \to (y \to x) \in B$, implies $(z \to y) \to (z \to x) \in B$, by Theorem 2.4 (c) B is a Hilbert ideal, this finished the proof.

References

[1] Jie Meng, Young Bae Jun. BCK-algebras. [M].Seoul: K YUNG MOON SA CO. 1994: 64~71.

[2] D. Busneag. Hilbert algebras of fractions and maximal Hilbert algebras of quotients. Kobe [J]. Math. 1988(5): 161~172.

[3] LiuFang, LiJizu. Hilbert algebras is an anti-positive implicative BCK-algebras (Chinese). [J]. Shanxi college of mining and technology. 1997, 15(2): 214~217.

[4] J. Ahsan, A.B. Thaheem. On ideals in BCK-algebras, Math. Seminar Notes, 1977(5): 167-172.

[5] W. H. Cornish. On positive implicative BCK-algebras, Math. Seminar Notes, 1980(8): 455-468.

978-1-935068-06-8 © 2009 SciRes.

Hierarchical and Dynamic Trusted Evaluation Model Based on Agent

YANG Xiaohui[1,2], ZHOU Xuehai[1], TIAN Junfeng[2]

1. School of Computer Science and Technology, University of Science and Technology of China, Hefei, China

2. Institute of Network Technology, Hebei University, Baoding, China

e-mail: yxh@hbu.edu.cn

Abstract: In order to improve current dynamic evaluation theories and methods of software, a hierarchical and trusted architecture based on agents is proposed, an autonomous and cooperative multi-agent system is implemented with reactive agent architecture, and the system trusted chain is expanded to the agents with the hierarchical trust expanding mechanism. The new idea of behavior trace and checkpoint scene is also proposed, a prospective behavior feature extraction mechanism of software and an actual behavior feature extraction mechanism are implemented base on the new idea, and a dynamic trusted evaluation model of software base on behavioral semantic distance is constructed.

Keywords: agent; dynamic trusted evaluation; behavior trace; checkpoint scene; behavioral semantic distance

基于 Agent 的层次化动态可信评测模型

杨晓晖[1,2]，周学海[1]，田俊峰[2]

1.中国科学技术大学计算机科学与技术学院，合肥，中国，230027

2.河北大学网络技术研究所，保定，中国，071002

e-mail: yxh@hbu.edu.cn

【摘　要】针对目前软件动态可信性度量理论和方法存在的问题，提出了一种基于 Agent 的层次化可信系统架构，采用反应式体系结构实现了自治协同的多 Agent 系统，并通过层次化信任扩展机制使系统信任链延伸到了 Agent 层；提出了行为轨迹和检查点场景的概念，并基于此实现了软件预期行为特征提取和软件实际行为监控机制，构建了基于行为语义距离度量的软件动态可信评测模型。

【关键词】agent；动态可信评测；行为轨迹；检查点场景；行为语义距离

1 引言

二十一世纪是信息的时代，信息技术和产业高速发展，空前繁荣，信息已成为一种重要的战略资源。但危害信息安全的事件层出不穷，形势非常严峻。信息安全事关国家安全和社会稳定，保障信息安全意义重大。

实践表明，大多数安全隐患来自于微机终端，要确保源头微机的信息安全，必须从微机的芯片、硬件结构、操作系统以及应用软件等方面综合采取措施。由此促成了可信计算技术的出现和发展[1]。

可信计算的思想源于社会，其基本思想为：首先构建一个信任根，再建立一条信任链，从信任根开始到硬件平台，到操作系统，再到应用软件，一级认证一级，一级信任一级，把这种信任扩展到整个计算机系统，从而确保整个计算机系统的可信[2]。

目前，可信计算已经成为信息安全领域的研究热点之一，可信计算技术与产品不断涌现。人们已经认识到，可信计算技术是一种行之有效的信息安全技术。与普通计算机相比，可信计算机的安全性大大提高。但可信计算机也不是百分之百安全，可信计算的发展尚存一些问题[1]：

（1）目前国内外在可信计算领域都处于理论研究滞后于技术开发的状况。至今，尚没有公认的可信计算理论模型。可信测量是可信计算的基础，但是目前尚缺少软件动态可信性的度量方法与理论。

（2）目前的可信测量只是系统开机时的系统资源静态完整性测量，因此只能确保系统开机时的系统资源静态完整性。这不是系统工作后的动态可信测量，因此尚不能确保系统工作后的动态可信性。

资助信息：国家自然科学基金项目(60873203)，国家 863 计划项目(2008AA01Z101).

针对上述问题，提出一种基于 Agent 的层次化可信系统架构，并基于软件行为特征提取和行为语义距离度量建立动态可信评测模型。

2 层次化可信系统架构 MMA

“MMA”是基于 Agent 技术的层次化可信系统架构“管理 Agent-监控 Agent-分析 Agent”（Manager agent-Monitor agent-Analyzer agent）的简称。MMA 系统架构如图 1 所示。

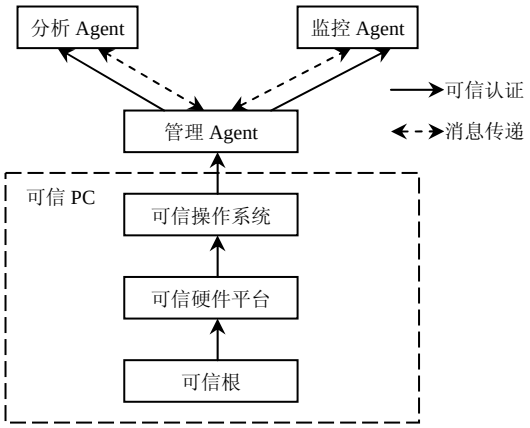

Figure 1. MMA: Hierarchical and trusted architecture

图 1. MMA：层次化可信系统架构

其中，分析 Agent 负责对待评测软件进行静态分析和动态跟踪，从中提取预期行为特征并上报管理 Agent；监控 Agent 负责软件运行时的实际行为监控和特征提取，并将所提取的实际行为特征上报管理 Agent；管理 Agent 负责提供系统与用户之间进行交互的用户界面，接受用户提交的待评测软件，计算软件实际行为特征与预期行为特征之间的行为语义距离，实现软件的动态可信评测，并向用户反馈评测结果。

2.1 反应式 Agent 体系结构

Agent 体系架构总体上分为审慎式、反应式和混合式三大类。在 MMA 架构中，Agent 的实现采用反应式体系架构，每个 Agent 都是由事件处理器 EP（Event Processor）、方法集 MS（Method Set）和状态集 SS（State Set）三部分构成。

Agent=<EP, MS, SS>

事件处理器 EP 主要负责 Agent 的行为控制，在 Agent 的生命期内始终持续自主地工作着。在 MMA 架构中，事件是与 Agent 有特定关联的特殊状态，如服务请求到达、特定状态被修改、警戒阈值被超越等。事件处理器可进一步细分为事件感知器（Event Sensor）、事件适配器（Event Adapter）和事件分发器（Event Dispatcher）三部分。

EP=<ES, EA, ED>

其中，事件感知器时刻捕捉其所关注的事件状态的出现，并根据事件状态的类型启动相应的事件适配器工作；事件适配器获取相关事件信息作识别，并将识别结果提交给相应的事件分发器，启动有关的事件处理方法执行。

方法集 MS 用来描述 Agent 处理相关事件的方法，体现了 Agent 的事件处理能力。分发集中各个方法的执行由事件分发器引发，在其执行过程中可能会影响 Agent 的内部状态，从而导致新的事件发生。

状态集 SS 用来表征 Agent 当前的各种属性和状态信息。在 Agent 的行为过程中，随着各种事情的发生，其状态可能会不断发生变化。

2.2 自治协同的多代理系统

MMA 架构所采用的反应式 Agent 的事件处理机制为实现各类 Agent 之间以及 Agent 与外部环境之间主动灵活的交互机制奠定了基础。

管理 Agent、监控 Agent 和分析 Agent 之间通过消息事件的处理实现了消息传递机制。三类不同的 Agent 可以驻留在相同的或者不同的主机上，通过基于规范的协调机制来实现 Agent 间的自治协同。每个 Agent 自主地对其监控的事件、拥有的方法、所处的状态进行合理安排，以调整各自的决策和行为，最大程度地实现各自目标。

基于规范的协调机制能够很好地处理和过滤恶意 Agent 的不良行为，同时还不违背 Agent 的自主性。这样，三类 Agent 既独立自主，又协调一致，从而实现一个分工合理明确、运行高效安全的自治协同的多代理系统 MAS。

2.3 层次化信任扩展机制

MMA 架构是一个层次化的可信系统架构，它将传统的系统信任链从软硬件系统延伸到三类代理中，这一过程是通过可信启动链的延伸来实现的。

在可信启动链建立的过程中，所有的信任都从 BIOS 中可信根的一段固定不变的可信代码开始，在把

978-1-935068-06-8 © 2009 SciRes.

系统控制权交给下一段代码之前，这段可信代码会去度量下一段将要执行的代码，并把度量结果扩展到可信 PC 的平台配置寄存器 PCR 中。如果每一段新的代码在移交系统控制权之前都去度量下一段代码的话，就可以建立起信任链。

如图 1 所示，从可信 PC 的可信根开始建立可信启动，先认证硬件平台，然后认证操作系统，再认证管理 Agent，最后由管理 Agent 认证监控 Agent 和分析 Agent，一级认证一级，一级信任一级，将可信启动链延伸到 Agent，从而实现 MMA 系统架构的层次化信任扩展机制，保障 MMA 系统架构的可信性。

3 动态可信评测机制

在 TCG 规范所描述的信任链扩展过程中，可信计算机采用的是基于数据的完整性测量机制，只能确保在系统开机启动的初始阶段系统软硬件资源的静态完整性。虽然系统的安全性能有所提高，但当系统开始运行后，操作系统和应用软件的行为却不能保证会按照预期的方式，朝着预期的目标运行，即目前的可信计算机无法对可能带来严重隐患和安全威胁的软件的动态行为进行可信测量和监控[3]。但是在信任链扩展过程中，系统软件和应用软件行为的可信以及它们之间的可信传递是不可逾越的，动态可信是可信计算亟待解决的关键问题之一[3,4]。

可信计算组织 TCG 从实体行为角度对可信进行了定义：如果一个实体的行为，总是以预期的方式，达到预期的目标，则称其为可信的[2]。武汉大学张焕国教授[3]指出：“可信≈可靠+安全”，即计算机系统所提供的服务是可靠的、可用的，信息和行为是安全的。屈延文教授[5]认为：软件是人类为了完成某类特定功能，在预期的软硬件环境下由信息处理装置对软硬件资源进行调度的一系列资源调度策略；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；主体的可信性是指主体行为的历史记录反映其是否违规、越权以及超出范围等方面的一种统计特性。

因此，要确保软件行为“总是以预期的方式达到预期的目标”，就必须针对软件的动态行为进行可信评测。

3.1 软件行为及其描述

定义 1. 软件行为 SB（Software Behavior）是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，都可以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。

$$SB=<BT, CS>$$

定义 2. 软件主体在给定的时间范围内从事的行为，按照时间顺序记录下来形成的形式化序列，称为该软件主体的**行为轨迹** BT（Behavior Trace）。

软件的执行过程是由控制流和数据流组成的，而 API 函数调用序列则是一个软件的具体行为表现。因此对软件行为轨迹的描述可以通过追踪软件的 API 函数调用序列来加以实现。

$$BT=<API_1, API_2, \ldots, API_n>$$

定义 3. 对给定的软件行为轨迹在某一时间点进行采样，取得的环境参数集合，称为行为轨迹在该时刻的**检查点场景** CS（Checkpoint Scene）。

$$CS=<PARA_1, PARA_2, ..., PARA_m>$$

3.2 软件行为特征提取与监控

3.2.1 软件预期行为特征提取

软件预期行为特征提取功能由分析 Agent 实现，包括预期行为轨迹提取和预期检查点场景提取。

预期行为轨迹提取主要采用控制流静态分析的方法，在不执行待评测软件的情况下，首先通过逆向工程获得软件的目标程序，然后通过对目标程序的可执行语句的若干可执行路径进行控制流分析，确定目标程序的控制结构，从而推断软件在动态执行时的预期行为，进而从中提取软件预期行为特征。

控制流分析分为过程内分析和过程间分析。过程内控制流分析可以确定子程序内语句的执行顺序，通过构建控制流图 CFG（Control Flow Graph）实现。控制流图中，节点表示指令或指令片段，节点之间的有向边表示指令间的跳转关系。为了降低 CFG 处理的复杂性，先将目标程序划分为基本块，然后用节点表示程序基本块，再用有向边连接基本块来创建 CFG。过程间控制流分析是确定子程序之间可能的控制流路径的过程，用调用图 CG（Call Graph）表示。调用图中的节点与子程序对应，节点之间的有向边则用来表示子程序间的调用关系。

CFG 和 CG 的存储均采用邻接表的数据结构，在空间效率和处理性能之间取得了一个较好的平衡。

预期检查点场景提取主要是基于数据流图实现简单的静态数据流分析，也是在不执行待评测软件的情

况下，收集目标程序数据的运行时信息，分析目标程序中的数据定义、数据使用以及数据对象之间的依赖关系。通过数据流分析，可以获取很多抽象层次要求较低的信息，对确定待评测软件的逻辑组成及其交互关系极为重要。

3.2.2 软件实际行为监控

软件实际行为监控由监控 Agent 实现，采用程序植入和系统监控等动态分析技术，在全局范围内更改目标程序代码以添加额外的监控操作，然后利用操作系统级虚拟化技术搭建一个沙盒系统，以实现对软件运行状况的虚拟动态追踪，通过多次执行待评测软件来尽最大努力获取目标程序的每一个可能的行为轨迹分支和检查点场景，从而充分获取程序的行为属性数据，实现对软件实际行为的监控和特征提取。

动态分析依赖于目标程序运行时的输入，不同的输入可能会得到不同的动态分析结果。对输入的依赖导致了动态分析的不完全，但同时也把待评测软件的输入、输出和软件行为紧密地联系起来。通过动态分析可以看到，程序输入的变化可以直接引起目标程序的内部行为和程序输出发生变化，从而导致监控 Agent 触发不同的事件。

为了更加全面、准确地刻画软件的行为特征，在动态分析的同时计算相邻检查点之间的执行时间间隔并以时间戳的方式记录为检查点场景的属性之一。

3.3 软件行为语义距离度量

分析 Agent 所采用的软件预期行为特征静态分析方法可以穷举目标程序所有可能执行路径，这样在监控 Agent 的沙盒系统中虚拟运行的待评测软件一旦偏离这些可能的执行路径，即可判定其行为不再可信。

为了精确描述实际行为与预期行为的偏离程度，通过将语义距离的概念和相关理论引入软件行为可信度量机制来实现软件的动态可信评测。

在安全可信的运行环境下，分析 Agent 提取并处理待评测软件的行为轨迹和检查点场景，获取该软件的预期行为特征；然后监控 Agent 提取并处理待评测软件在未知可信的运行环境下的行为轨迹和检查点场景，获取该软件的实际行为特征；最后通过度量实际行为与预期行为之间的偏差，并将其转化为行为语义距离，从而实现对软件行为的动态可信评测。

语义距离（Semantic Distance）通常基于相似性（Similarity）或相关性（Relation）度量函数来实现语义距离度量。为实现软件行为语义距离度量机制，引入“行为语义距离”对软件行为之间的相关度、偏离度和相似度进行定量描述和计算。定义行为轨迹偏离函数、检查点场景相似函数、行为语义距离度量函数，根据动态追踪获取的软件实际行为轨迹和检查点场景与分析得到的预期行为轨迹和检查点场景之间的相关程度和偏离程度，计算软件实际行为与预期行为之间的语义距离。

4 相关研究

目前软件动态可信评测已经取得了一些研究成果，相关研究主要包括：

4.1 系统调用序列

Stephanie Forrest 等人提出通过监视特权进程的系统调用序列的入侵检测模型[6]，通过系统调用短序列来刻画进程的特征。南京大学姚立红等提出的 CTBIDS 检测模型[7]利用系统调用特征树描述程序行为特征，通过异常有限积累或者海明距离判别程序入侵。南开大学贾春福教授等[8]根据系统调用的作用效果对系统调用进行划分。西安交通大学冯力等[9]通过监控主机上的系统调用序列为观察对象建立预测模型，基于规划识别理论提出了一种带参数补偿的贝叶斯网络动态更新算法。中国科学技术大学谭小彬等[10]利用隐马尔可夫模型来描述特权进程正常运行时局部系统调用之间存在的规律性。北京交通大学田新广等[11]利用数据挖掘技术中的序列模式对特权程序的正常行为进行建模，根据系统调用序列的支持度和可信度在训练数据中提取正常模式。

4.2 行为可信

沈昌祥院士提出一个基于系统行为和可信状态的多级安全模型 MSMBTS[12]，以 BLP 模型为基础，引入可信度和可信状态测量函数，利用可信计算平台进行基于系统行为的完整性测量、存储和报告。国防科技大学王怀民教授[13]提出，行为可信是研究如何通过对协同计算的软件行为进行约束，避免个体软件的行为损害系统目标，从而提高软件群体的可信性。

4.3 软件属性及其规范描述

Borut Jereb[14]使用属性及其规格来描述软件，并将属性分为四类：任务系统、软硬件及数据环境、时

978-1-935068-06-8 © 2009 SciRes.

间限制和执行历史。Mark Dredze 等人[15]使用基于信用衡量的线性分类器来对软件行为进行分类。Marius Kloft 等人[16]使用支持向量数据描述 SVDD 来进行软件行为特征的描述和自动选择。

上述研究在行为建模、检测准确性、检测能力、系统实用性等方面仍有待进一步完善。

5 结束语

以瑞达公司的 JTQ-900 可信 PC 为硬件平台，在 Windows 2003 Server 操作系统上采用 Java 语言基于 JADE（Java Agent Development Framework）技术实现了基于 Agent 的 MMA 层次化可信系统架构和基于行为语义距离度量的软件动态可信评测机制，沙盒系统采用 Sandboxie 3.40。目前原型系统已经能够初步实现恶意代码的检测。

下一步的研究工作将在软件行为特征描述策略和基于多维主观逻辑的行为语义距离度量机制等方面继续展开。

6 致谢

本文的研究工作受到了国家自然科学基金项目（60873203）和国家 863 计划项目（2008AA01Z101）的资助，在此表示感谢。

References (参考文献)

[1] SHEN Chang-xiang, ZHANG Huan-guo, FENG Deng-guo, *et al.* Survey of Information Security [J]. Science in China Series F: Information Sciences, 2007, 37(2), P129-150 (Ch).
沈昌祥, 张焕国, 冯登国等. 信息安全综述[J]. 中国科学 E 辑: 信息科学, 2007, 37(2), P129-150.

[2] Trusted Computing Group. TCG Specification Architecture Overview [EB/OL]. https:// www. trustedcomputinggroup.org/ groups /TCG _1_0_Architecture_overview.pdf.

[3] ZHANG Huan-guo, LUO Jie, JIN Gang, *et al.* Development of Trusted Computing Research [J]. Journal of Wuhan University (Natural Science Edition), 2006, 52(5), P513-518 (Ch).
张焕国, 罗捷, 金刚, 等. 可信计算研究进展[J].武汉大学学报(理学版), 2006, 52(5), P513-518.

[4] PENG Guo-jun, ZHANG Huan-guo. Dynamic Trustiness Authentication Framework Based on Software's Behavior Integrity [C]. In: Proceedings of ICYCS2008, Zhangjiajie, Hunan, China. 2008, P2283-2288.

[5] QU Yan-wen. Software Behavior [M]. Beijing: China Publishing House of Electronics Industry (PHEI), 2005(Ch).
屈延文. 软件行为学[M]. 北京: 电子工业出版社, 2005.

[6] Forrest S, Hofmeyr S, Somayaji A, *et al.* A sense of self for unix processes [C]. In: Proceedings of the 1996 IEEE Symposium on Security and Privacy, IEEE Computer Society, Washington, DC, USA, 1996, P120-128.

[7] YAO Li-hong, ZI Xiao-chao, XIE Li, *et al.* Research of System Call Based Intrusion Detection [J]. Chinese Journal of Electronics, 2003, 31(8), P1134-1137 (Ch).
姚立红, 訾小超, 谢立, 等. 基于系统调用特征的入侵检测研究[J]. 电子学报, 2003, 31(8), P1134-1137.

[8] JIA Chun-fu, ZHONG An-ming, ZHOU Xia, *et al.* Research on Syscall-based Intrusion Detection Technology for Linux System [J]. Application Research of Computers, 2007, 24(4), P147-150 (Ch).
贾春福, 钟安鸣, 周霞, 等. 基于系统调用的 Linux 系统入侵检测技术研究[J]. 计算机应用研究, 2007, 24(4), P147-150.

[9] FENG Li, GUAN Xiao-hong, GUO San-gang, *et al.* Plan Recognition Based Method for Predicting Intrusion Intentions of System Call Sequences [J]. Chinese Journal of Computers, 2004, 27(8), P1083-1091 (Ch).
冯力, 管晓宏, 郭三刚, 等. 采用规划识别理论预测系统调用序列中的入侵企图[J]. 计算机学报, 2004, 27(8), P1083-1091.

[10] TAN Xiao-bin, WANG Wei-ping, XI Hong-sheng, *et al.* A Hidden Markov Model Used in Intrusion Detection [J]. Journal of Computer Research and Development, 2003, 40(2), P245-250 (Ch).
谭小彬, 王卫平, 奚宏生, 等. 计算机系统入侵检测的隐马尔可夫模型[J]. 计算机研究与发展, 2003, 40(2), P245-250.

[11] TIAN Xin-guang, QIU Zhi-ming, LI Wen-fa, *et al.* Anomaly Detection of Program Behaviors Based on System Calls and Data Mining [J]. Computer Engineering, 2008, 34(2), P1-3 (Ch).
田新广, 邱志明, 李文法, 等. 基于系统调用和数据挖掘的程序行为异常检测[J]. 计算机工程, 2008, 34(2), P1-3.

[12] ZHANG Xiao-fei XU Fang SHEN Chang-xiang. Research on Multilevel Security Model Based on Trustworthy State and Its Application [J]. Chinese Journal of Electronics, 2007, 35(8), P1511-1515 (Ch).
张晓菲, 许访, 沈昌祥. 基于可信状态的多级安全模型及其应用研究[J]. 电子学报.

[13] WANG Huai-min, TANG Yang-bin, YI Gang, . Trusted Mechanism of Internet Software [J]. Science in China Series F: Information Sciences, 2006, 36(10), P1156-1169 (Ch).
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: 信息科学, 2006, 36(10), P1156-1169.

[14] Borut Jereb. Software describing attributes [EB/OL]. Computer Standards & Interfaces, 2008, doi:10.1016/j.csi.2008.06.012.

[15] Mark Dredze, Koby Crammer, Fernando Pereira. Confidence-Weighted Linear Classification [C]. In: Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 2008, P264-271.

[16] Marius Kloft, Ulf Brefeld, Patrick Düssel, et al. Automatic Feature Selection for Anomaly Detection [C]. In: Proceedings of AISec2008, Alexandria, Virginia, USA, 2008, P71-76.