A Feather-Weight Application Isolation Model*
(Work-in-Progress)

Jianhua Liao¹, Yong Zhao², and Changxiang Shen²

¹ School of Electronics Engineering and Computer Science,
  Peking University, Beijing, China
  liao_jh@139.com
² Department of Computer Science and Technology,
  Beijing University of Technology, Beijing, China
  zhaoyonge_mail@sina.com

Abstract. In this paper, we introduce a new application isolation model which bases on Least-Privilege principle and Need-to-Know principle. Since this model is easy to implement, we call it the Feather-weight Application Isolation (FAI) model. This model is used to achieve the Process Permission Constraint (PPC) and classified Object Access Control (OAC). The model allows us to make application isolation depending on PPC policies and OAC policies. Compared with the existing complex isolation models such as sandboxes and virtual machines, the FAI model is simpler, and therefore it does not only meet the necessary security requirements but also increases the usability. To isolate applications and prevent classified objects of the applications from being illegally tampered, the FAI model extends the traditional two-dimensional access control matrix to a three-dimensional access control matrix, which includes subjects, objects and processes. In order to support multi-level security and Mandatory Access Control (MAC), the concept of processes sensitivity level ranges is considered in the model. In this article, we first give an informal description of the model, and then introduce the formalized description and safety analysis. Finally we explain the feasibility of the model by showing the result of the engineering implementation.

Keywords: Security model, Application isolation, Access control, Process constraint.

1 Introduction

Currently, applications of computers are getting more powerful and more complex than before such as grid computing [1]. And sometimes, many application systems are deployed on the same platform to reduce hardware cost. The ideal situation is that application system only accesses its own data, but reality is not the case. In many situations a user probably operates several application systems at the same time, and then in the

* This project was funded by the Chinese National 863 Plan (Project code: 2009AA01Z437) and 973 Plan (Project code: 2007CB311100).
multi-application systems coexisting situation, there must be some interferes among those application systems because of user's misoperation, mutual confusion system data and so on. The result will definitely affects application system's security.

Application system software itself is getting more and more complex. It will inevitably have some harmful bugs such as Buffer overflows [2]. Hackers, viruses and Trojan horses [2] can take advantage of those security vulnerabilities to compromise computer systems and promote their privileges, and then to access sensitive information or control other processes, and furthermore threat integrity and confidentiality of the system.

The security problems of application systems do not just originate from software itself, but also from malicious software (malware) such as viruses, worms, Trojan horses and so forth. It is inevitable to have malwares in the platform because of widely used network and more and more complex application environment. Furthermore, malwares detection problem is undecidable [3]. Malwares always infiltrate or damage computer systems, and then badly threat security of application systems.

The basic reasons of those security problems which be mentioned above are confusion of application environment, fuzzy application boundary. The implicit sharing that exists in modern operating systems is another source of threaten which can result in leaking sensitivity information. There are no strict policies to protect sensitive information. Then the result is that there must be some interferes among application systems. So the most effective way to resolve those security problems is application isolation [4]. This is also the key point of this article.

To resolve those problems mentioned above and meet the fault-tolerant and intrusion-tolerant requirements of computer systems [5] [6], we introduce an isolation model named *feather-weight application isolation* (FAI) model which bases on existing trust platform. The key techniques behind FAI model are process permission constraint and object access control, which are two-way control. Process permission constraint (PPC) is one aspect of FAI model which bases on the least privilege principle and follows the corresponding PPC policies. PPC constrains behaviors of the controlled-process and gives it the only permissions to carry out its normal functions. To all other unknown processes PPC give them only minimal permissions. We know that PPC can isolate processes but it can not isolate classified objects. So another assurance mechanism named object access control (OAC) is introduced which bases on need to know principle and makes only those processes according with OAC policies can access the objects. Those two aspects of FAI model isolate application-relative processes as well as protect application-relative sensitive objects, and they are the outstanding feature of FAI model.

## 2 Relative Works

Researches on application isolation mainly include noninterference model of information flow, sandbox and virtual machine.

### 2.1 Noninterference Information Flow Model

The concept of noninterference theory model is first proposed by Goguen and Meseguer [7]. From then on there are many information flow models which base on