Ell Secure Information System Using Modal Logic Technique

Yun Bai, University of Western Sydney, Australia
Khaled M. Khan, Qatar University, Qatar

ABSTRACT

In this paper, the authors propose a formal logic technique to protect information systems. As the widespread use of computer systems grows, the security of the information stored in such systems has become more important. As a security mechanism, authorization or access control ensures that all accesses to the system resources occur exclusively according to the access polices and rules specified by the system security agent. Authorization specification has been widely studied and a variety of approaches have been investigated. The authors propose a formal language with modal logic to specify the system security policies. The authors also provide the reasoning in response to system access requests, especially in situations where the security agent’s knowledge base is incomplete. The semantics of this language is provided by translating it into epistemic logic program in which knowledge related modal operators are employed to represent agents’ knowledge in reasoning. The authors demonstrate how this approach handles the situation where the security agent’s knowledge on access decision is incomplete. The proposed mechanism effectively prevents unauthorized and malicious access to information systems.

Keywords: Access Control, Authorization, Formal Language, Intelligent Systems, Modal Logic, Reasoning, System Security

1. INTRODUCTION

As the widespread use of computer systems, the security of the information stored in such systems has become more and more important. When measuring computer system performance, the system security is just as important as its efficiency, speed and cost. There are several mechanisms to provide computer system security. Typical ones are authentication and authorization. The authentication service only allows legitimate users access the system, while the authorization service controls the legitimate users only performing legitimate operations on the system resource. Authorization is to ensure that all accesses to the system resources occur exclusively according to the access polices and rules specified by the security agent of the information system. In this paper, we investigate the authorization or access control mechanism. Authorizations or access control has been widely studied (Atluri et al., 2002; Chomicki et al., 2000; Fernandez et al., 1995; Zhou et al., 2008) and a variety of authorization specification approaches such as access matrix (Dacier et al., 1994; Denning, 1976), role-based access control (Crampton et al., 2008), access control in

DOI: 10.4018/jsse.2011040104
database systems (Bertino et al., 1996; Meadow, 1991), authorization delegation (Murray et al., 2008), procedural and logical specifications (Bai et al., 2003; Bertino et al., 2003) have been investigated. Since logic based approaches provide a powerful expressiveness (Fagin et al., 1995; Das, 1992; Fernandez et al., 1989) as well as flexibility for capturing a variety of system security requirements, increasing work has been focusing on this aspect. Jajodia et al. (2001) proposed a logic language for expressing authorizations. They used predicates and rules to specify the authorizations; their work mainly emphasizes the representation and evaluation of authorizations. The work of Bertino et al. (2000) describes an authorization mechanism based on a logic formalism. It mainly investigates the access control rules and their derivations. In their recent work (Bertino et al., 2003), a formal approach based on C-Datalog language is presented for reasoning about access control models. Damiani et al. (2002) presented a language for specification of access control by exploiting the characteristics of XML to define and enforce access control directly on the structure and content of the document. They provided a flexible security mechanism for protecting XML documents. Murata et al. (2003) proposed a static analysis for XML access control. Given an access control policy and an access query, they use a static analysis to decide if to grant or deny such an access request. In this way, run-time evaluation is only needed when the static analysis is unable to decide. This pre-execution analysis improves the performance of the system response to a query.

However, there were some limitations so far in these approaches. These approaches made access decision based on the security domain the agent know, and the domain is complete. When the security agent does not have complete, specific information about the security domain, how to reason and answer access queries under such a scenario? For instance, if the agent knows that Alice can access the classified file if Alice is a manager. At the moment, it is believed that Alice is a manager. Is Alice allowed to access the file?

Another example, the agent currently does not know clearly who can access the classified file between Alice and Bob, but knows only one of them can. This can be specified by a disjunctive logic program (Baral, 2003) as follows:

\[
\text{AliceCanAccessFile} \lor \text{BobCanAccessFile} \leftarrow,
\]

\[
\text{AliceCanAccessFile} \leftarrow \text{not BobCanAccessFile},
\]

\[
\text{BobCanAccessFile} \leftarrow \text{not AliceCanAccessFile}.
\]

If a query asks if Alice can read the classified file, the agent will not be able to make the decision, because this program has two different answer sets: \{AliceCanAccessFile\} and \{BobCanAccessFile\}. In fact, under many circumstances, using disjunctive logic programming to specify security policies is not sufficient to precisely handle incomplete information.

In this paper, we propose a knowledge based formal languages \( \mathcal{L}^k \) to specify authorization domain with incomplete information in secure computer systems. We introduce modal logic to specify and reason about a security domain then translate the domain into an epistemic logic program. We show that our approach has an expressive power to describe a variety of complex security scenarios.

In our presentation, we assume the existence of a single, local system security officer or security agent administering the authorizations. This assumption enables us to concentrate on a single administering agent system and hence avoids the problem of coordination among multi agents. Access control in a multi-security system or administered by multi-security agents is out of the scope of this paper.

The rest of the paper is organized as follows. Section 2 describes language \( \mathcal{L}^k \) by outlining its syntax and gives some authorization policy examples specified by the language. Section 3 explains the semantics of language \( \mathcal{L}^k \). We start by introducing a general overview of epistemic logic program, then map the domain description specified by \( \mathcal{L}^k \) into the logic program, and we give some examples to show the
Related Content

Reduction of Defect Misclassification of Electronic Board Using Multiple SVM Classifiers
www.igi-global.com/article/reduction-of-defect-misclassification-of-electronic-board-using-multiple-svm-classifiers/111448?camid=4v1a
Specifying Knowledge Graph with Data Graph, Information Graph, Knowledge Graph, and Wisdom Graph
[www.igi-global.com/article/specifying-knowledge-graph-with-data-graph-information-graph-knowledge-graph-and-wisdom-graph/201482?camid=4v1a](www.igi-global.com/article/specifying-knowledge-graph-with-data-graph-information-graph-knowledge-graph-and-wisdom-graph/201482?camid=4v1a)

Vulnerabilities of Secure and Reliable Low-Power Embedded Systems and Their Analysis Methods: A Comprehensive Study
[www.igi-global.com/chapter/vulnerabilities-of-secure-and-reliable-low-power-embedded-systems-and-their-analysis-methods/116106?camid=4v1a](www.igi-global.com/chapter/vulnerabilities-of-secure-and-reliable-low-power-embedded-systems-and-their-analysis-methods/116106?camid=4v1a)

Teaching Software Architecture in Industrial and Academic Contexts: Similarities and Differences
[www.igi-global.com/chapter/teaching-software-architecture-in-industrial-and-academic-contexts/188205?camid=4v1a](www.igi-global.com/chapter/teaching-software-architecture-in-industrial-and-academic-contexts/188205?camid=4v1a)