Merging Heterogeneous Security Orderings

P.A. Bonatti¹, M.L. Sapino¹ and V.S. Subrahmanian²*

¹ Università di Torino
{bonatti,mlsapino}@di.unito.it
² University of Maryland
vs@cs.umd.edu

Abstract. The problem of integrating multiple heterogeneous legacy databases is an important problem. Many papers [7, 9, 3] to date on this topic have assumed that all the databases comprising a mediated/federated system share the same security ordering. This assumption is often not true as the databases may have been developed independently by different agencies at different points in time. In this paper, we present techniques by which we may merge multiple security orderings into a single unified ordering that preserves the security relationships between orderings. We present a logic programming based approach, as well as a graph theoretical approach to this problem.

Keywords: Theoretical Foundations of Security, Heterogeneous Mediated/Federated Systems.

1 Introduction

Complex applications in today's rapidly changing world require the ability to access a wide variety of distributed, heterogeneous data sources. In order to assist the author of such complex applications, Wiederhold [14, 15] has proposed the important concept of a mediator as a paradigm for integrating heterogeneous data and software. Major efforts towards the construction of mediators are currently underway at many universities and companies [12, 1, 4, 10].

One of the fundamental problems faced by each and every one of these efforts is the problem of security. Besides the standard problems related to security in databases and information systems, one has to tackle specific problems raised by the heterogeneous nature of data sources. In particular, it is not reasonable to assume that all packages participating in a mediated system and the mediator itself use the same security orderings.

* Partially supported by the Army Research Office under grant DAAH-04-95-10174, by the Air Force Office of Scientific Research under grant F49620-93-1-0065, by ARPA/Rome Labs contract Nr. F30602-93-C-0241 (Order Nr. A716), by NSF Young Investigator award IRI-93-57756, by NSF award IRI-93-14905, and by the Army Research Laboratory under Cooperative Agreement DAAL01-96-2-0002 Federated Laboratory ATIRP Consortium.
For example, a Supply Mediator might use an inventory relation, created by an Army Logistics agency, and a GIS database, developed by a defense contractor, and it may very well be the case that these two groups used different security orderings in their design. A mediator must be able to automatically combine the security orderings used by the different packages in a semantically meaningful and security-preserving way. Such a combination will result in a new security level ordering that can be used by the mediator. In this paper, we will present methods by which different security orderings used by individual packages may be neatly combined into a new security ordering that captures the essential properties of the constituent orderings. Furthermore, we will show that this combination may be elegantly implemented using well known logic programming techniques and graph algorithms, and is solvable in polynomial time.

In the next section, we introduce the basic principles of our approach to the combination of heterogeneous security orderings. In Sec. 3, we introduce an axiomatization of the combinability problem based on a logic program $P$, and show how a global security ordering can be computed by querying $P$. In Sec. 4 we introduce an alternative approach, based on a graph representation of the problem, which leads to more efficient algorithms. We conclude with a section on related work.

2 Combining Heterogeneous Security Orderings

A security ordering is a partial order $(S, \leq)$, where $S$ is a set of security levels and $\leq$ is a partial order over $S$. Intuitively, in an individual data source, each user is assigned a security level that determines the user’s capabilities; higher security levels correspond to higher clearance. Informally speaking, we would like to construct a security ordering $(S, \leq)$ that takes into account and semantically merges the security orderings $(S_1, \leq_1), \ldots, (S_z, \leq_z)$ that are used by the different packages participating in a mediated system. Users may then be assigned security levels from the merged ordering $(S, \leq)$.

Example 1. Figure 1 shows two different security orderings used by two relational DBs, $db1$ and $db2$. The first database uses the ordering $u$ (unclassified), $s$ (secret), $ts$ (top-secret), $ts-sci1$, and $ts-sci2$, where $ts-sci$ stands for top-secret special compartmented information. The classification levels of $db2$ may be read similarly. For now, the reader can ignore the dotted line in Figure 1. There are several things to note, however. As $db1$ and $db2$ may have been created independently, it is entirely possible that these orderings refer to different things. For example, it may well be the case that the classification level $ts$ in $(S_1, \leq_1)$ is equal to the classification level $s$ in $(S_2, \leq_2)$. Later (Example 2) we will show how this may be captured within our framework.

The aim of this section is to find a way of combining a given set of security orderings $(S_1, \leq_1), \ldots, (S_z, \leq_z)$. In our framework, we will merge the given security orderings into a new ordering in such a way that certain constraints are preserved. These constraints express relationships between security levels in