A System to Specify and Manage Multipolicy Access Control Models

Elisa Bertino
DSI
Università degli Studi di Milano
bertino@dsi.unimi.it

Barbara Catania
DISI
Università degli Studi di Genova
catania@disi.unige.it

Elena Ferrari
DSCFM
Università degli Studi dell’Insubria
Elena.Ferrari@uninsubria.it

Paolo Perlasca
DSI
Università degli Studi di Milano
perlasca@dsi.unimi.it
Summary

• The general problem: Data Security
• MACS: a multipolicy access control system
• The architecture of MACS
• How MACS works
• Conclusions and future work
Data Security

- Data are an important strategic and operational asset for any organization
- Damages and misuses of data affect not only a single user or an application; they may have disastrous consequences on the entire organization.

Data must be protected!
Data Security

Data Security requires:

- **Confidentiality**
- **Integrity**
- **Availability**
Data Security

- A comprehensive solution for Data Security consists of:
  - the identification of the security requirements
  - the specification of a security policy
  - the selection of some mechanism to enforce the specified policy
Data Security: Access Control System

• An *access control system* regulates the operations that can be executed on data and resources to be protected
  – an *access control policy* can be enforced through a set of authorization rules, establishing the operations and rights that subjects can exercise on the protected objects
  – the *reference monitor* determines whether an access requests can be authorized or not, according to the authorization rules enforcing the selected policy
Issues in Data Security

- A variety of access control policies have been so far defined
- Articulated access control requirements are not adequately supported by a single-policy access control mechanism
What MACS is?

• MACS is a multipolicy access control system supporting both the specification and the implementation of a large variety of access control models
What MACS is?

• MACS is *flexible* and *extensible* since:
  – it can easily accommodate new access control requirements
  – it allows the administrator to define its own access control policies and/or models in addition to those already provided by the system
What MACS is?

• MACS is based on a formal language and provides a set of tools helping the administrator in the specification and analysis of access control models and authorization management.
How?

• Under MACS multiple access control policies can co-exist within the same system.

• The basic idea is to apply different policies to different disjoint sets of the objects to be protected.
How?

• An access request involving an object o is authorized or denied according to the policy enforced by the specific component model containing o
MACS: the language

- MACS is based on the C-Datalog language which is an Object-Oriented extension of Datalog.
- C-Datalog supports:
  - classical object-oriented concepts, such as classes, objects and inheritance (used to represent subjects, objects, privileges, sessions, …)
  - typical logic-based concepts, such as deductive rules (used to represent authorization and constraint rules)
MACS: the language

- Each instance of an ACM is a logical program composed of C-Datalog rules defined against a C-Datalog schema specifying the structure of the elements existing in the system
MACS: the architecture

• The architecture of MACS consists of two main environments with different tasks:
  – the *Multipolicy Management Environment (MME)*:
    • Generation of a template (a template partially specifies the components belonging to each instance of a model)
    • Static analysis of the generated template
  – the *Run-Time Environment (RTE)*:
    • Generation of an authorization base according to the template
    • Verification of end-user and SA requests
MACS: the template

• A multipolicy template specifies, for each component model:
  – the set of objects to be protected by this component model
  – a set of data and rules representing the structural components on which the model is based
  – a set of rules establishing how authorizations are derived and propagated along the hierarchical organization of the structural component
  – a set of rules specifying integrity constraints
  – a conflict resolution function to deal with conflicting authorizations
RBAC Component Model

On
R1
R2
R3
R4

User-role Assignments
Constraints (SSD, DSD)
P

Permission-role Assignments

Partially Specified Model

Fully Specified Model

User

RBAC Component Model

Permissions-role Assignments

Constraints (SSD, DSD)

Partially Specified Model

Fully Specified Model
MACS: the MME environment

- The main modules of the MME environment are:
  - the Graphic Template Interface (GTI)
  - the Static Analyzer

- The Graphic Template Interface supports the PA during the generation of a template, whereas the Static Analyzer checks consistency of the generated template.
The MME performs the following tasks:

- **Generation of a template**
- **Static analysis of the generated template**

A. Partitioning of protected objects  
B. Assignment of an ACM to each partition  
C. Assignment of a CRF to each ACM  
D. Generation of the template  
E. Analysis of generated template  
F. Feedback analysis answers  
G. Run-time environment
MACS: the Run-time environment

• The main modules of the Run-Time Environment are:
  – the Authorization Manager Front-End
  – the Access Control Compiler and Checker
  – the Authorization Analyzer

• The Authorization Manager Front-End manages end-user and SA requests
MACS: the AC compiler & checker

- The main tasks of the Access Control Compiler and Checker are:
  - the generation of an Authorization Base according to the policies stated by the PA
  - the verification of end-user and SA requests

- The Authorization Analyzer:
  - supports the compiler during the generation process of an Authorization Base and
  - checks consistency and correctness of the generated set of authorizations
The Run-Time Environment performs the following tasks:

- **Generation of an authorization base**
- **Verification of End-user and SA requests**

  a. Complete the template
  b. Send the template instance to the compiler
  c. Check consistency and correctness
  d. Send a feedback to SA
  e. Generate a consistent AB
  f. Submit SA requests
  g. Generate a consistent Authorization Base
  h. Submit End-user requests
MACS: how it works

- **An Access Control Model Schema (ACMS)** defines the structural components on which the model is based.
- **Access Control Model Instance (ACMI)** provides information concerning the component instances, that is, the “actual” subjects, objects, privileges and sessions, and the authorizations and constraint rules used to instantiate the model.
MACS: how it works

• The components of an ACMI can be organized as follows:
  – Domain classes represent the structure of the basic components (s, o, p, and sessions) of an ACM, whereas domain instances represent the actual components (instances are represented as set of facts)
  – Domain structure information represents relationships existing between basic components
  – The authorization component contains a set of facts and rules representing direct authorizations
  – The propagation component contains a set of rules by which additional authorizations can be derived
  – The constraint component consists of a set of rules specifying static and dynamic constraints on the basic components
MACS: how it works

• An *Access Control Multipolicy Template (ACMpt)* of arity n is a set of n tuples such that:
  – each of them is a partially specified ACMI handling a specific partition of objects to be protected and solving conflicts according to a specific conflict resolution function

• When all ACMIs are fully specified, we refer to them as an *Access Control Multipolicy Template Instance (ACMptI)*
MACS: how it works

• The *semantics* of an ACMpTI is the union of the semantics of its component ACMPs, whereas an *authorization base* is the consistent set of authorizations specified by its semantics.
Conclusions

• We have presented MACS, a flexible multipolicy access control system, supporting the specification and the analysis of a large variety of ACMs

• It is based on a logical language providing a formal basis for the development of advanced analysis tools

• The prototype implementation is based on Eclipse
Future work

• We are formally studying the problem of authorization and template analysis
  – we are also investigating the computational cost of this analysis

• We plan to support multiple policies for the same objects