Meca: A Tool for Access Control Models

Amal Haddad

LSR-IMAG, Grenoble, France
Amal.Haddad@imag.fr

1 Introduction of Meca

Access Control is a technique which insures security by preserving confidentiality and integrity of information. Meca (Models for access control) is a tool which generates, in a B machine, operational conditions that should be verified by an application to insure security.

The inputs of Meca are a B machine offering a format for presenting a security model and a functional model containing the presentation of an application with its sensitive entities like variables and operations. The format of a security model provides a declarative representation of the access distribution in the system at a given moment. It is done according to various models related to three branches policies: discretionary policies model (DAC)[4], Bell and LaPadula model (BLP) [1], Biba model (Biba) [2] and role based access control model (RBAC) [3]. Meca generates access rules in a B machine called security kernel. Security kernel offers secure services under witch sensitive entities of functional model can be manipulated. The format of the security kernel varies depending of the security policy model type.

In access control scope, Objects are passive entities that represent system resources and should be protected. Subjects are active entities accessing to objects and possessing rights to manipulate them. Figure 1 presents Meca with his inputs and outputs components.

We illustrate our approach and Meca with a small part of a bank card example.

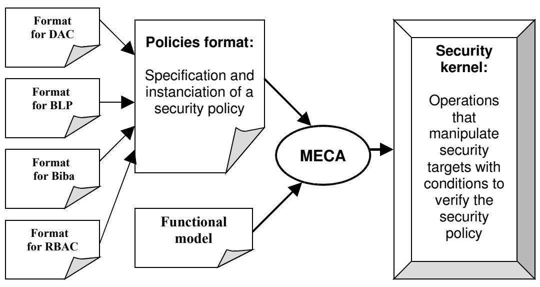
Format for DAC
Format for BLP
Format for Biba
Format for RBAC
Policies format:
Specification and instanciation of a security policy
Functional model
MECA
Security kernel:
Operations that manipulate security targets with conditions to verify the security policy

Fig. 1. MECA schema

2 Use Case and the Functional Model

The bank card is a smart card with an electronic purse. As the other smart cards [5], the purse can be debited from a terminal in a shop to pay a purchase and credited at an ATM from a cash or by withdrawing from a bank account. There are three different hierarchical levels to operate the purse: debiting the purse, crediting the purse from cash or from a bank account and performing administrative operations. With these three access levels correspond three kind of terminals from which the purse can be manipulated: a terminal in a shop for debiting the purse, a bank terminal for crediting the purse and an administration terminal for updating configuration parameters. We note that any operation allowed on a specific level is allowed on lower levels.

During its life cycle, the card passes through different operating modes load, use or invalid. When the card is in the load mode, its issuer (the bank) records the holder pin code (hpc). We present the specification of SetHPC operation in the functional model:

```
MACHINE BankCard
SETS
  MODE={load,invalid,use}
CONSTANTS
  HPC
PROPERTIES
  HPC = 0000 .. 9999 /*pin has four digits*/
ABSTRACT_VARIABLES
  mode,hpc
INVARIANT
  mode ∈ MODE ∧ hpc ∈ HPC /*variable hpc is typed in the invariant*/
INITIALISATION
  hpc :=1000 || mode:= load
OPERATIONS
  setHPC(pin) =
      PRE
          pin ∈ INT ∧ pin ∈ HPC ∧ mode=load
      THEN
          hpc := pin
      END
END
```

Thereafter, we present the second entry component for Meca, it is the format for the security model.

3 Policies Format

As a part of our work, we deal with three models of security policies: The access matrix model (belonging to DAC policies), Bell Lapadula model (BLP), Biba model (belonging to MAC policies) and Role Based Model (RBAC). We will give a short definition about each of these policies. In the scope of discretionary policies (DAC), access control on objects is granted to their creator. Access permissions are read and write [4]. Mandatory policies (MAC) provide a classification for subjects and objects. According to this classification, read and write operations are allowed to preserve secrecy (BLP model) [1] or integrity of information (Biba model) [2]. The originality of role based access control policies (RBAC) is that they introduce role concept that intercepts permissions, subjects and objects [3]. Permissions are granted to roles. Subjects obtain permissions according to roles that they hold. Permissions are operations that could be executed on objects. Roles inherit between each others permissions and users [6]. We will give a detailed account of RBAC instanciated with bank card example. The aim is to safeguard security when the card is manipulated from various terminals.

```
MACHINE rbac_format_BankCard
SETS
  SUJET ={shop_terminal,bank_terminal,admin_terminal};
  ROLE ={debit,credit,admin };
  PERMISSION ={OP_ checkHPC,OP_setHPC ,OP_ debitPurse,
  OP_creditPurse}
CONSTANTS
  sujet_role, role_permission, herite_de
PROPERTIES
   sujet_role ∈ SUJET → ROLE
   ∧ role_permission ∈ ROLE ↔ PERMISSION
   ∧ herite_de ∈ ROLE ↔ ROLE
   ∧ closure1 ( herite_de ) ∩ id ( ROLE ) = ∅
   ∧ sujet_role = {(shop_terminal ↦ debit),
  (bank_terminal ↦ credit),(admin_terminal ↦ admin)}
   ∧ role_permission = {(admin ↦ OP_setHPC),
  (credit ↦ OP_creditPurse),(debit ↦ OP_debitPurse),
  (debit ↦ OP_checkHPC)}
   ∧ herite_de = {(admin ↦ credit),(credit ↦ debit)}
END
```

The entities represented in bold are fixed by Meca format while those in simple depend on the application. A commentary introduced in the header component announces for Meca the security model used.

The set PERMISSION refers to operations of the application. Subjects are terminals. Roles are various levels that exist. The relation sujet_role is the association between a subject and a role. In this example we suppose that a subject can have only one role. Role_permission is the association between a role and one or several permissions. The hierarchy is presented by the relation called herite_de. We use constants to present these sets since we don´t deal with updating access distribution. The prefix OP_ is used for operations in order to avoid names clash with functional component.

4 The Generated Security Kernel

The security kernel is the component that contains conditions that should be verified to insure the respect of security. This kernel is produced according to access rules that govern each model type of security policy. Thereafter, we present security kernel for RBAC format.

Meca generates for every operation, that belongs to set PERMISSION of RBAC format, a new predicate to control access. SetHPC operation becomes:

```
setHPC ($pin$,$su$)=
  PRE
    $pin \in \mathbf{INT} \wedge pin \in HPC \wedge mode = invalid$
     $\wedge\ su \in SUJET$
     $\wedge\ (sujet\_role(su), OP\_setHPC):(\mathbf{closure}(herite\_de);$
    $role\_permission)$
  THEN
    $hpc := pin$
  END
```

This mechanism reinforces the precondition of each operation that should be performed by a specific role. The generated component should be used instead of the old functional model. Hence, whenever SetHPC is called its precondition should be satisfied.

5 Contribution and Future Works

Our contribution is materialised by the approach to introduce security in the development of certified products. The security kernel could be used in the scope of a static verification (by B proofs) to reason about the security of an application. In a sooner future work, we will exploit the use of the security kernel for a validation with tests. In the scope of Meca, we formalize also model for security policies. We are now interested to develop object oriented security models and to formalize it in the B method. This is done in the scope of POSE project (http:/wwwrntl-pose.info/).

References

1. D. E. Bell and L. J. Lapadula. Secure computer systems: A mathematical model. Technical report esd-tr-278, vol. 2, The Mitre Corporation, Bedford, 1973.
2. K. J. Biba. Integrity considerations for secure computer systems. Technical report tr-3153, The Mitre Corporation, Bedford, 1997.
3. D. F. Ferraiolo and D. R. Kuhn. Role based access control. In *15th National Computer Security Conference*, 1992.
4. Butler W. Lampson. Protection. In *5th Princeton Symposium on Information Science and Systems*, pages 437–443, 1971.
5. Renaud Marlet and Cédric Mesnil. A demonstrative electronic purse -Card specification-. pages 1–53, 2002.
6. R.S. Sandhu. Role hierarchies and constraints for lattice-bases access controls. *Lecture Notes in Computer Science*, 1146:65–79, 1996.