

SPIDERNet: the Security Policy Derivation for Networks Tool

F. Barrère, A. Benzekri, F. Grasset, R. Laborde[♦], B. Nasser,
Université Paul Sabatier - IRIT/SIERA -
118 Rte de Narbonne F31062 Toulouse Cedex04
Telephone: +33 (0) 5 61 55 60 86 - Telecopy: +33 (0) 5 61 52 14 58
{barrere,benzekri,grasset,laborde,nasser}@irit.fr

Abstract:

The construction of a security policy becomes increasingly difficult because of the complexity of the factors to consider. As long as networks were composed of closed infrastructures, security relied on models based on the end-systems access control. Today, if these models still remain typical, they must be completed by the definition and usage of "security models for networks". The design of such models must allow to approach the security concerns in a pragmatic way rather than relying exclusively on the expertise of the network administrator. For this objective, we raise the question : when do we implement network security mechanisms? After shedding light on the weaknesses of the access control models, we show the strong tie between the policy defined by the system administrator and that defined by the network administrator. Then we define modelling tools and rules that start from the users rights to determine constraints on the filtering rules and encryption mechanisms to be used. A prototype called SPIDERNet implemented in PROLOG language validates our approach.

Keywords: *Secure Networking, Security Policy, Security Verification.*

1. Introduction

The networks and more particularly the Internet continue to bring new working means and tools which are not risk free on these networks' previewed usage. Even though the enterprises are conscious of the importance of e-commerce and e-business technologies around, they are also aware of the security problems caused by opening their information systems to internetworking. Each technology used, each facility brought to the users proves to be a new challenge for the networks administrators who must control the data flows in their communication infrastructure and for the system administrators who must control the accesses to the resources. What should be their security policy? Is it correct? A security policy is the regulation that governs the behavior of the system within a security context. From a technical point of view, the behaviour of a device is a function of the configurations set up on it. So, one of the major problems of policy implementation is to determine these configurations starting from a fixed objective.

Traditionally the Access Control Models (ACMs) were used to define and put in place a security policy [14]. The system is modelled in term of objects and subjects. The policy aims at indicating the access rights of the subjects on the objects (for example users' rights on files in UNIX systems). These models were defined within the operating systems and databases framework; they were employed thereafter in networks. ACMs differ by the manner of treating the authority which defines the policy (Discretionary Access Control [8], Mandatory Access Control [2,3], hierarchical organisation [1,12], cooperative organisation [4]) and the way of approaching the policy targets (classification by trellis [2,3], role approach [11]).

Nevertheless, ACMs offer powerful tools for controlling fixed data, i.e., that doesn't travel across a network. In fact, data goes through a chain of devices. Depending on the network topology, these devices may be heterogeneous (terminal devices, routers, proxies, authentication servers). For example, data confidentiality within a group of persons can be guaranteed by access control mechanisms set up at the device containing these data, access control of the data flows by firewalls, ALGs (Application Level Gateway), encryption protocols as PGP, SSL, IPsec, L2TP. The problem is to decide which devices interfere in the realisation of a stated security objective. It is necessary to check the consistency of the existing configurations. Contrary to databases or to operating systems access control problem, few works were interested to check if a security objective is applied within the network context (checking filtering configuration [6], IPsec configuration [7]).

We propose then a tool providing recommendations on the configurations to be laid down, starting from the definition of the users' rights. In the first part we expose our solution. We choose a method to specify the users' rights on data. Then we propose a notation to specify the abstraction of the network topology. With these elements, we define rules and properties to permit determine recommendations of the security mechanisms to be implemented, starting from the specification of the users' rights and the network topology. In the second part, we describe our implementation of the above concepts in PROLOG language. Finally, we show an example of a VPN using our prototype.

[♦] Author to contact for more information

2. Towards an end-to-end security policy¹

If the security modelling proposals can be considered as satisfactory solutions for particular problems, it is no more the case considering the global view. Since:

- The Access Control models ignore the security problems concerning the communication infrastructure to concentrate on that concerning data access,
- The networks world ignores the access control policy applied on information.

However, in a distributed environment the access control policy of users to data or devices and the security policy of network elements are correlated. In fact, why protecting a flow if it doesn't contain restricted access information. To manage the security at the network infrastructure level, we should take into account the access control policy (example 1). Moreover, contrary to the network security policies, the control access policies are independent of the topology of the communication infrastructure.

Our initiative to derive a policy consists in:

- 1) Defining the access control policy to resources for the users,
- 2) Deducing from these policies the security policy of the network according to its topology.

Example 1:

Consider a CSMA (Carrier Sense Multiple Access) type network (Figure 1) where three devices are connected:

- Device A is used by a person having the Read/Write rights on data found in server C
- Device B is used by a person having no access right to these data.

According to the protocol operation (CSMA), each message sent on the network is broadcasted to all, so it is necessary to implement security mechanisms to protect the communications between device A and server C. The policy for securing the flows depends directly on the access control as well as the network infrastructure topology.

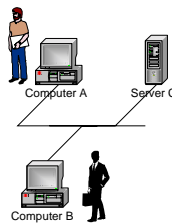


Figure 1 : Example of a local network

After choosing an access control policy model, we propose a method for modelling the topology of a network and finally rules allowing to determine if it is or not necessary to employ security mechanisms at the network level for a given an access control policy and a topology.

2.1 The choice of an access control model

It is important to call back that it is the enterprise needs that determine the security requirements, the models should thus permit to formulate the security needs compared to the enterprise operation. The RBAC [1, 11] fit very well in this context because it proposes to formulate the rights of the users according to their place in the enterprise. It consists then in determining "roles", where a role corresponds to a set of privileges and responsibilities associated with a particular activity in the enterprise. We prefer then the role based policies RBAC on other models because the latter do not take into consideration the notions of group and responsibility which are two important concepts for an enterprise structure.

Moreover [1] proposes a calculus for the RBAC:

- A says s : A makes the statement s,
- $A \Rightarrow B$: A is a stronger principal than B ; for example if B represents a group then A may represent a member or a subgroup of B,
- A serves B : A is a delegate for B,
- $A|B$: (A|B) says s iff A says B says s,
- A for B : A speaks on behalf of B with appropriate delegation certificates, i.e., (A|B) says s and A serves B,
- A as R : A in role R.

¹ We borrow this term from the QoS world to show the need to consider the users and network devices in the security policy.

Example 2: Consider the simple case of Figure 2:

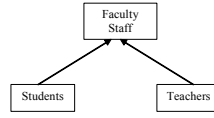


Figure 2 : Example of a roles hierarchy

We have 3 groups: Faculty Staff, Students and Teachers. In addition, we have the relations $Students \Rightarrow Faculty Staff$ and $Teachers \Rightarrow Faculty Staff$. If we consider a PhD student P who is lecturer too then, $P \Rightarrow Students$ and $P \Rightarrow Teachers$. Consequently, P can use the roles $R_{students}$ and $R_{teachers}$; this is noted by $P as R_{students}$ and $P as R_{teachers}$.

2.2 Modeling the network topology

We reuse the information flow control approach [13,5] to model the communication infrastructure which binds the users and the data. We take as a hypothesis that the implementations are sure, by Common Criteria or ITSEC evaluation. Our process consists of modelling security functionalities and interactions between these functionalities (Figure 3). For example, a stateless firewall has the Filter functionality.

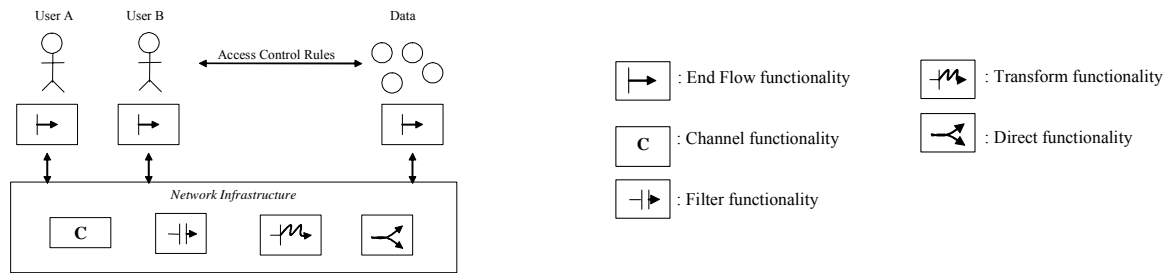


Figure 3 : Modelling of network topology

In the following we will consider the relations:

- The active_Object and passive_Object relations to indicate that an entity is active or passive,
- The active_EF and passive_EF relations to indicate that an EndFlow functionality is active or passive,
- The relation Is_Connected permits to define if there is a connection between two entities.

2.2.1 End Flow functionality

An End Flow (EF) is a functionality specific to the terminal devices, i.e., the servers (data and application) and the workstations, PC ... It transmits flows between the applications/users and the network. We consider two types of End Flow functionalities:

- **Active End Flow functionality (AEF):** An EF is said active if any active entity (for instance, users or subjects in BLP model [2]) is connected to this EF.

Let EF be the set of End Flow functionalities and O be the set of objects in access control model,

$$\forall ef \in EF, \forall s \in O, active_Object(s) \wedge is_connected(s, ef) \Rightarrow active_EF(ef)$$

- **Passive End Flow functionality (PEF):** An EF is said passive if any passive entity (all that is not active, for example, objects in BLP model) is connected to this EF.

Let EF be the set of End Flow functionalities and O be the set of objects in the access control model,

$$\forall ef \in EF, \forall o \in O, passive_Object(o) \wedge is_connected(o, ef) \Rightarrow passive_EF(ef)$$

2.2.2 Channel functionality

The Channel functionality models the physical network. It receives the flow on an interface and retransmits it to all entities connected to this channel. This functionality may be viewed as the bus architecture (i.e. CSMA). When a flow is oriented, it is not only received by the addressed destinations but by all of the systems connected to this channel (Cf example 1).

2.2.3 Filter functionality

The Filter functionality stops or not a data flow. We find this functionality in firewalls, Application Level Gateways, filtering routers...

2.2.4 Transform functionality

The Transform functionality (TR) accepts a flow "f" as input and according to the associated management rules, it transforms this flow to another flow "f' ". This new flow may have properties such as confidentiality, integrity and authenticity ensuring a certain security level.

2.2.5 Direct functionality

This functionality determines the path that a data flow will follow.

For more clearness in the following, we show only the used functionalities. Figure 4 shows how the example of Figure 1 can be modelled. The devices A, B and C are represented by End Flow functionalities EF A, EF B and EF C. The Ethernet network is modelled by the channel C1.

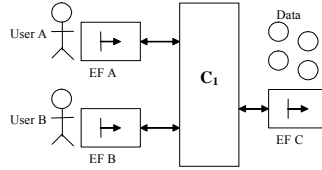


Figure 4 : Example 1 modelling

2.3 Relation between application level and network level policies

There is no magical recipe to define the security policy for a given network. Indeed, for the same security objective multiple solutions can be considered. Securing the data transferred can be achieved using different security protocols (PPTP, L2TP, IPsec, SSL, TLS, PCT, Secure HTTP, PGP, S/MIME, SSH) following the needs, blocking access to the network (1,2,3 level VLAN for Ethernet networks, possibilities offered by IP routing, filtering), etc. We propose a method that allows, starting from an RBAC type access control policy to define recommendations on what security means to be implemented.

Let us consider a simple example given by the groups of Figure 2. The topology is described in Figure 4. The data of End Flow EF C contains the file grouping the final exams. The users with role Teachers_Role have the rights to read and write in the Exam_Doc file. Logically, the users with role Students_Role do not have these rights.

2.3.1 An RBAC based policy specification

Let the following RBAC policy defined in ASL [9] be associated with our example:

```

/* We associate user A with the Teachers group and user B with the Students group */
rel(userA,Teachers)
rel(userB,Students)
/* Those with role Teachers_Role have the Read/Write rights on Exam-Doc */
cando(Exam_Doc, Teachers_Role,+read)
cando(Exam_Doc,Teachers_Role,+write)
/* Definition of the organisation of the enterprise in Figure 2 */
dirin(Teachers, Faculty Staff)
dirin(Students, Faculty Staff)
/* Assigning roles to the groups */
rel(Teachers, Teachers_Role)
rel(Students, Students_Role)
/* Rules that permit to derive the role rights for a user */
dercando(object, subject, action) : cando(object, role, action), rel(subject, group), in(group, group'),
rel(group', role)
/* Applying rights, our policy is closed2 */
do(object, subject, action) : dercando(object,subject,action)
do(object, subject, - action) : not dercando(object,subject,action)
  
```

As for the way the channel is specified, when user A reads the Exam_Doc document, the user B can access the document while crossing the network. The policy goal is then not fulfilled. This error comes up because the channel is not trusted by the members of the Teachers group. Let us define what a trusted channel is.

2.3.2 Trust property of a channel

In the following we will consider two relations:

- The Role relation provides a group or a user with the associated role,
- The Privilege relation provides a given role with the set of associated privileges.

Role transmission property:

Let EF be the set of EndFlow and $U = \{ \forall s \in O, \text{active_Object}(s) \}$
 $\forall ef \in EF, u \in U, \text{Is_Connected}(u,ef) \Rightarrow \text{Role}(u) = \text{Role}(ef)$

Proof: Using the calculus of [1], let U be a user, G_R be a group, and R be the role associated with group G_R . If “ $U \Rightarrow G_R$ ” then U can use the role G so “U as R”. A user who connects at an EndFlow is noted by “EF|U”. We have then “EF|(U as R)”. From the associative property of these functions we obtain “(EF|U) as R”. Consequently when a user U of the group G_R connects to the EndFlow EF with the role R, EF has also the role R.

² In a closed policy, what is not explicitly permitted is prohibited.

Let now consider two groups, G1 associated with the role R1 and G2 with R2. If the groups G1 and G2 have concurrent activities, their privileges are consequently different, each one desires to indirectly obtain rights of the other group. There may be thus a conflict of interest between two groups. In the same way, the group that has more privileges than the other is in conflict with the latter, but not the inverse. We express this by the definition of the property of conflict of interests between roles.

Definition of conflict of interests among roles:

Let $S_1 = \text{Privilege}(R_1)$ and $S_2 = \text{Privilege}(R_2)$.

We say that a role R_1 is in conflict of interest with role R_2 iff $\neg(S_1 \subseteq S_2)$.

We define then the relation $\text{Is-In-Conflict} : \text{role} * \text{role} \rightarrow \text{Boolean}$. This relation is:

- Non reflexive : $\neg(R_1 \text{ Is-In-Conflict } R_1)$
- Non symmetric : $(R_1 \text{ Is-In-Conflict } R_2) \neq (R_2 \text{ Is-In-Conflict } R_1)$
In fact, let R_1 and R_2 such that $S_1 \subset S_2$ then $\neg(R_1 \text{ Is-In-Conflict } R_2)$ and $(R_2 \text{ Is-In-Conflict } R_1)$.
- Intransitive : $\neg((R_1 \text{ Is-In-Conflict } R_2) \wedge (R_2 \text{ Is-In-Conflict } R_3) \text{ implies } (R_1 \text{ Is-In-Conflict } R_3))$
For example, let R_1, R_2 and R_3 be such that $S_1 \subset S_3$ and $S_2 \cap S_3 = \emptyset$,
 $(R_1 \text{ Is-In-Conflict } R_2)$, $(R_2 \text{ Is-In-Conflict } R_3)$ but $\neg(R_1 \text{ Is-In-Conflict } R_3)$ because $S_1 \subset S_3$.

We define the special role Enemy that represents the group of attackers. This role has no privileges, then: Let R be the set of roles, $\forall r \in R, r \text{ Is-In-Conflict Enemy}$.

Moreover, a group is never in conflict with any of its subgroups. Indeed, if $G \Rightarrow F$, then

$\text{Privileges}(\text{role}(F)) \subseteq \text{Privileges}(\text{role}(G))$, then $\neg(\text{Role}(F) \text{ Is-In-Conflict Role}(G))$.

The fact that a role R_1 is in conflict of interest with R_2 puts constraints on the events between the corresponding EndFlows. Consequently, due to the modelling of the Channel functionality, the role transmission property and the property of conflicts among roles, we can define the trusted channel property.

Trusted channel property:

Let C be the set of channels and R be the set of roles,

A channel $c \in C$ is said to be trusted for a role $r \in R$ and noted $\text{trust}(c,r)$ iff

$\forall ef \in EF, \text{active_EF}(ef) \wedge (r' = \text{Role}(ef)) \wedge \text{is_connected}(ef, c), \neg r' \text{ is_in_conflict } r'$

For instance, in our example we have $\neg \text{trust}(C_1, \text{Teachers_Role})$ and $\text{trust}(C_1, \text{Students_Role})$.

2.3.3 When to implement an encryption mechanism?

If a channel is not trusted, the information traversing it is subject to attacks. However, not all exposed information needs to be protected, for example an email telling about holidays. Consequently the implementation of a security mechanism depends on the information itself. We assign two tags to data: C to indicate that it needs to be confidential, and I for integrity protection needs. By defining the trusted channel, we thus set two rules for the security mechanisms in a network:

- Confidentiality rule: It is necessary to apply confidentiality mechanisms to data flow “d” for a role “r” on a channel “c” if the channel “c” is not trusted for “r” and “d” has the tag C .
- Integrity rule: It is necessary to apply integrity mechanisms on a data flow “d” for a role “r” on a channel “c” if the channel “c” is not trusted for “r” and “d” has the tag I .

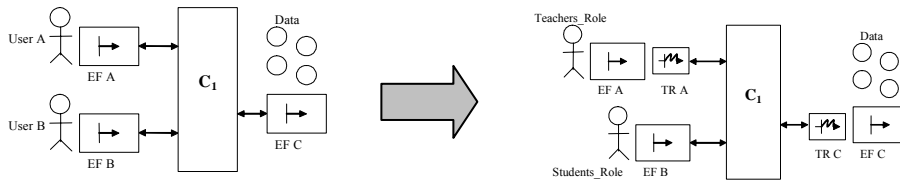


Figure 5 : Securing our example

We set also a last rule:

Confidentiality mechanisms \Rightarrow Integrity mechanisms \Rightarrow Authentication mechanisms

This rule upholds since if we prohibit a user from reading a file, logically the access for writing will be prohibited too. While if we prohibit a user from writing into a file, that does not imply obligatory the prohibition of the user from reading it. Moreover, setting up control mechanisms with respect to a user and/or a device has no sense except when guarantying the identity of this user and/or device. Consider again the preceding example, the file Exam_Doc is confidential and its integrity should be preserved because it contains the final exams. We add then the tags C and I to this file. Moreover: *Teachers_Role Is-In-Conflict Students_Role* then the channel is not trusted. Consequently, the data flow between the two EndFlows EF_A and EF_C requires confidentiality and integrity mechanisms as shown in Figure 5.

We do not identify any more a user by his name but by his role. This is because the role determines the trust property of the channel. This makes it possible to incorporate all the users who have the same role and thus to reduce the specification overhead.

We note messages exchange by using [1]:

- TR_A for EF_A for userA as Teachers_Role says m' signifies that TR_A has modified the message $m(C,I)$ coming from EF_A and sent by userA in a flow m' .
- TR_C for EF_C for Data as Teachers_Role says m' signifies that TR_C has modified the message $m(C,I)$ coming from EF_C and sent by the data server in a flow m' .

Where $m' = \{m(C,I)\}_{K_i,K_c}$ where K_i represents the protection concerning integrity, and K_c protection concerning confidentiality.

The message m' corresponds to message m transformed by TR_A and TR_C . The message m' has no more the tags C and I because the new message doesn't need to be protected. The operator *for* states implicitly that the identities are guaranteed. It is useful to choose this notation since it allows to easily represent the constraint on the Transform functionality. For example TR for EF for U as Role says $\{m(C,I)\}_{K_i,K_c}$ gives, in the case of IPsec protocol [10], information for configuration:

- “ TR ” corresponds to the entity implementing IPsec mechanisms,
- “(EF for U as Role)” corresponds to the entities having the right to send and receive messages via IPsec security mechanisms, for example the configuration of the SPD (Security Policy Database),
- “ $\{m(C,I)\}_{K_i,K_c}$ ” for the security mechanisms to be set up. Thus the configuration of the SAD (Security Association Database), in our example K_c implies the use of ESP protocol.

Now, we have a tool allowing to determine if a data flow must be protected or not on a given channel. It remains to implement another tool to decide if a flow should be blocked.

2.3.4 When to implement a filtering rule?

The Filter functionality receives a flow then it allows it to pass or not. Firewalls represent such a functionality. Let us analyse now how these devices are configured. First, the firewalls are classically represented with an IN and an OUT interfaces. The IN interface corresponds to the enterprise network one, where the OUT interface to the Internet. We may conclude that IN is associated with trusted networks where OUT with the inverse. Now let us examine why an administrator would decide to block certain flows while letting others pass.

The OUT to IN flow:

A firewall will block a flow from OUT towards IN to prevent it modifying the state of the system connected to IN. That is preventing that the information in this flow write into the system. For example, a filtering rule may be set to avoid that an attacker raids on a system server. Thus the filtering rules from OUT to IN try to guarantee the system integrity. We may establish this design by applying the Biba model [3].

The IN to OUT flow:

A firewall will block a flow from IN to OUT to prevent information from leaving the system to outside. The goal is then to avoid that the information reach directly (by a program of spyware type or a non-trusted user) or indirectly (because of an awkwardness) to someone who hasn't the authorization to get it. So, the filtering rules from IN to OUT permit guarantying the confidentiality of the system's information. This result is shown with the BLP model [2].

The system IN is trusted while the OUT is not. From the definition of the relation $Trust(c,r)$, if a channel is trusted then it is IN, else it is OUT. We thus release the following rules:

- A Filter functionality between two channels $C1$ and $C2$ such that $Trust(C1,R)$ and $Trust(C2,R)$, doesn't block any flow for the role R ,
- A Filter functionality between two channels $C1$ and $C2$ such that $\neg Trust(C1,R)$ and $\neg Trust(C2,R)$, doesn't block any flow for the role R . If a message sent by R has any tag C or I , it is secured because of confidentiality and integrity rules,
- A Filter functionality between two channels $C1$ and $C2$ such that $Trust(C1,R)$ and $\neg Trust(C2,R)$, $C1$ represents the IN interface and $C2$ the OUT interface:
 - If message m has the tag C of the form X for ... for Z as R says $m(C)$ and it is sent from IN to OUT, then it should be blocked,
 - If message m has at least the tag I of the form X for ... for Z as R says $m(I)$ and it is sent from OUT to IN then it should be blocked.

3. Implementation: SPIDERNET

Based on the properties defined above, we have implemented a prototype called SPIDERNET using PROLOG language. It takes as an input topology specification and an RBAC policy specification and deduces through two algorithms where Transform functionalities must be installed and what Filter functionalities must do.

3.1 “Adding security to an EndFlow functionality” Algorithm

Let EF be the set of EndFlow functionalities, TR be the set of Transform functionalities, C be the set of Channel functionalities.

```

For all  $ef$  in  $EF$  do
  Let  $R = \{Role(ef)\}$ 
  For all  $r$  in  $R$  do
    Let  $c \in C$  such as  $Is\_Connected(c,ef)$ 
    If  $trusted(c,r)$  Then
      If  $\exists tr \in TR$  such as  $Is\_Connected(ef,tr) \wedge Is\_Connected(tr,c)$  for the role  $r$  Then
        Remove the Transform functionality  $tr$  for the role  $r$ 
      EndIf
    Else
      Add a Transform functionality  $tr$  for the role  $r$  between  $ef$  and  $c$ 
    EndIf
  EndFor
EndFor

```

3.2 “Adding Security to a Filter functionality” algorithm

Let EF be the set of EndFlow functionalities, F be the set of Filter functionalities, TR be the set of Transform functionalities, C be the set of Channel functionalities and R be the set of roles.

```

For all  $f$  in  $F$  do
  Let  $c_1, c_2 \in C, c_1 \neq c_2$ , such as  $Is\_Connected(f,c_1) \wedge Is\_Connected(f,c_2)$ 
  For all  $r$  in  $R$  do
    If  $Trusted(r,c_1) \wedge Trusted(r,c_2)$  Then
      Add  $f$  permits data flow of the form “ $EF$  for  $X$  says  $m$  as  $r$ ”
      Remove  $f$  permits data flow of the form “ $TR$  for  $EF$  for  $X$  says  $m$  as  $r$ ”
    Else
      If  $\neg Trusted(r,c_1) \wedge \neg Trusted(r,c_2)$  Then
        Add  $f$  permits data flow of the form “ $TR$  for  $EF$  for  $X$  says  $m$  as  $r$ ”
        Remove  $f$  permits data flow of the form “ $EF$  for  $X$  says  $m$  as  $r$ ”
      Else
        If  $Trusted(r,c_1) \wedge \neg Trusted(r,c_2)$  Then  $IN \leftarrow c_1$  and  $OUT \leftarrow c_2$ 
        If  $\neg Trusted(r,c_1) \wedge Trusted(r,c_2)$  Then  $IN \leftarrow c_2$  and  $OUT \leftarrow c_1$ 
        Add a Transform functionality  $tr$  between  $IN$  and  $f$  for the role  $r$ 
        Add  $f$  permits data flow of the form “ $TR$  for  $EF$  for  $X$  says  $m$  as  $r$ ”
        Remove  $f$  permits data flow of the form “ $EF$  for  $X$  says  $m$  as  $r$ ”
      EndIf
    EndIf
  EndFor
EndFor

```

3.3 A VPN example

Let us consider a group named VPN which represents a collaborative group of engineers. These persons can identify themselves with the role VPNRole. Figure 6 represents the network topology specification defined as input and Figure 7 the result obtained with our prototype.

The policy associated is the following RBAC policy:

```

cando( $data_1$ , VPNRole, +read/write)
cando( $data_2$ , VPNRole, +read/write)
cando( $data_1$ , OtherRole, +read)

```

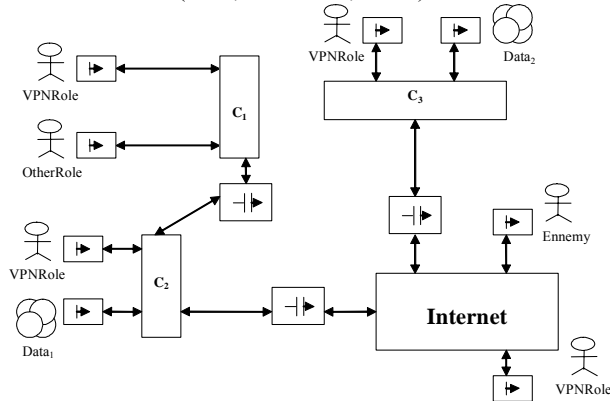


Figure 6 : Topology specification

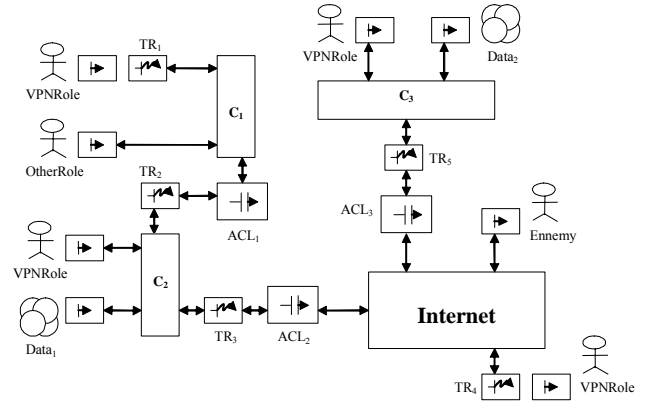


Figure 7 : Result obtained with our prototype

We get the relations: $VPNRole$ $Is_In_Conflict$ $OtherRole$ and $\neg OtherRole$ $Is_In_Conflict$ $VPNRole$
 Then: $\neg Trusted(c_1, VPNRole)$, $Trusted(c_1, OtherRole)$, $Trusted(c_2, VPNRole)$, $Trusted(c_2, OtherRole)$,

\neg Trusted(Internet,VPNRole), \neg Trusted(Internet,OtherRole), Trusted(c3,VPNRole).

Our prototype adds five Transform functionalities (Figure 7) such as: TR_1 transforms flows for VPNRole entities, TR_2 transforms flows for VPNRole entities, TR_3 transforms flows for VPNRole and OtherRole entities, TR_4 transforms flows for VPNRole entities, and TR_5 transforms flows for VPNRole entities. It also adds access controls on Filter functionalities:

- ACL_1 : Permit TR for EF for X says m as VPNRole, and Permit EF for X says m as OtherRole,
- ACL_2 : Permit TR for EF for X says m as VPNRole, and Permit TR for EF for X says m as OtherRole,
- ACL_3 : Permit TR for EF for X says m as VPNRole, and Permit TR for EF for X says m as OtherRole.

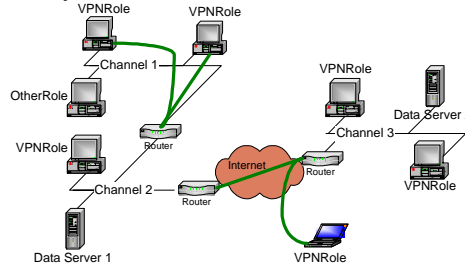


Figure 8 : An example of tunnel interconnection

In the case of the use of VPN technology, each Transform functionality for VPNRole corresponds to a tunnel end-point. We have to establish all the tunnels between all Transform functionalities for VPNRole; for instance $TR_1 \leftrightarrow TR_2$, $TR_3 \leftrightarrow TR_5$, $TR_4 \leftrightarrow TR_5$ (Figure 8).

4. Conclusion

In order to better understand the implementation of security policies, we have shown the lacks at the security modelling level in a world of information technology composed essentially of distributed applications. We proposed a notation allowing to specify the topology of a network while referring only to the interactions between security functionalities. Then we proposed a set of rules and properties implemented in a tool using PROLOG language that provide recommendations on the security mechanisms to be laid down at the network level, starting from the RBAC specification of user rights.

While many points have to be clarified, we will focus our future work on the following ones:

- The first one is concerning the problems associated when multiple roles are granted to the same user. It is highly probable that these roles be in conflict. How to take into account such a constraint?
- Our effort is oriented to a unique question: why to use security mechanisms? The relation *Is-In-Conflict* doesn't permit to take into account the importance of each data. Indeed, a role may get in conflict with another for a data D and not for another data D'. That is why we find it feasible to refine this relation.
- The third point concerns our analysis of the Filter functionality that considers only the information sender but not the destination. A user, who possesses rights on a particular file, has the right to reach the device on which the file is placed. Then we have to change our notation for message sending.

5. References

- [1] Abadi M., Burrows M., Lamson B., Plotkin G., "A Calculus for Access Control in Distributed Systems", in ACM Transactions on Programming Languages and Systems, vol 4, 706 -- 734, 1993.
- [2] Bell D., Lapadula L., "Secure Computer Systems: Mathematical foundations and model", MITRE corporation, Bedford M.A., 1973.
- [3] Biba K., "Integrity constraints for secure computer systems", USAF Electronic System Division, Bedford M.A., 1977.
- [4] Bonatti P., De Capitani di Vimercati S., and Samarati P., "An Algebra for Composing Access Control Policies" in ACM Transactions on Information and System Security, vol. 5, n. 1, February 2002.
- [5] Forcadi R., "Analysis and automatic detection of information flows in systems and networks", Technical Report UBLCS-99-16, University of Bologna, Italy, 1999.
- [6] Guttman J., "Filtering postures: Local enforcement for global policies", IEEE Symposium on Security and Privacy, Oakland CA, USA, 1997.
- [7] Guttman J., Herzog A., Thayer F., "Authentication and confidentiality via IPsec", 6th European Symposium in Computer Security ESORICS, Toulouse, France, 2000.
- [8] Harrison M., Ruzzo W., Ullman J., "Protection in operating systems", "Communication of the ACM", 1976.
- [9] Jajodia S., Samarati P., Sapino M.L., Subrahmanian V.S., "Flexible supporting for multiple access control policies" ACM Transaction on Database Syssem, 2000.
- [10] Kent, S., and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.
- [11] Lawrence G., "The role of roles", Computers and Security (12)15--21, 1993.
- [12] Lupu, E. "A Role-Based Framework for Distributed Systems Management", Ph.D. Dissertation, Imperial College, Department of Computing, London, U.K, July 1998.
- [13] Mantel H., "Information Flow Control and Applications - Bridging a Gap". FME 2001: Formal Methods for Increasing Software Productivity, International Symposium of Formal Methods Europe, Berlin, Germany, 2001.
- [14] Samarati P., De Capitani di Vimercati S., "Access Control: Policies, Models and Mechanisms", Foundations of Security Analysis and Design, R. Focardi and R. Gorrieri (eds), LNCS 2171, Springer-Verlag. 2001.