NETWORK SECURITY MANAGEMENT: A FORMAL EVALUATION TOOL BASED ON RBAC POLICIES

Romain Laborde, Bassem Nasser, Frédéric Grasset, François Barrère, Abdelmalek Benzekri
IRIT/SIERA Université Paul Sabatier, 118 Rte de Narbonne, F31062 Toulouse Cedex04 France

Abstract: The complexity of factors to consider makes increasingly difficult the design of network security policies. Network security management is by nature a distributed function supplied by the coordination of a variety of devices with different capabilities. Formal evaluation techniques should be used to ensure that correct security network strategy are enforced. In this paper, we propose a new formal tool which allows to describe a given network security strategy, a network topology and the security goals required. The tool includes an evaluation method that checks some security properties and provides information to refine the strategy used. We introduce an example of VPN architecture which validates our approach.

Key words: Policy, Network Security, Security Management, Security Evaluation

1. INTRODUCTION

Basically, the security of distributed applications is supported by a set of network security services which are implemented by means of security mechanisms. The security administrator should determine the *security services to use* and the *security mechanisms configurations to apply*. End to end security (e.g. SSL based solution) is often used, but it leads to conceal the underlying network. If such solutions can provide confidentiality, integrity, non repudation and authenticity properties, it is not suitable regarding the availability, anonymity property (e.g. deny of service, non accessibility) or regarding networks characteristics (throughput, flow control...). Then the design, the operation, and the maintenance of these

network configurations constitute an important part of the security management task.

Network management is by nature a distributed function supplying the coordination of a variety of devices with different capabilities (PC, firewall, secure gateways, routers, etc.). Once deployed, network security often become unmanageable over time since more rules are added and there is a real difficulty in retrieving, managing and getting rid of old unnecessary rules.

The first category of security management problems is the *security mechanisms inconsistency*. It can be divided into two sub-groups: the *atomic* security mechanisms inconsistency and the *distributed* security mechanisms inconsistency. The atomic inconsistency problem considers that two or more configuration rules on the same device can be incompatible. For example, one rule states that data flows with the source IP addresses in the range 10.0.0.0 can pass through the firewall and another rule on the same firewall states that the data flow with the source IP address 10.20.30.4 is denied. Several techniques[1] can be used to solve it, for example:

- *Denials take precedence* : negative authorizations take precedence,
- *Most specific take precedence* : the authorization that is most specific w.r.t. a partial order wins,
- *Positional*: the priority of the authorization depends on which they appear in the authorization list,
- *Priority level* : each authorization is assigned with a priority level, the authorization with the highest priority wins,
- Etc.

The distributed inconsistency concerns incompatible rules mapped on different devices. Thus, the administrator should pay a special attention to all dependency relations between rules present on different devices. For instance, an IPsec tunnel is correctly configured between two VPN gateways and a firewall between them blocks their IPsec data flows. Some works provide a partial solution considering only one kind of device, for example firewalls[2-4], IPsec gateways[5] or filtering IPsec gateways[6].

Nevertheless, security mechanisms consistency does not imply that the security objectives are achieved (i.e. the administrator has chosen the good security services). The latest management paradigms[7-9] aim to automate the management tasks. In this context, policy based management approach[7,10,11] considers abstract security policies that can represented at different levels[12,13] ranging from the business goals to the devices-specific configurations. The process that transforms a definite goal into the corresponding configurations is called derivation process[14]. Thus, it tries to define the relation between the objectives and mechanisms configuration.

As there is no formal and automatic evaluation method of the couple services/mechanisms against the objectives yet, we are working on the definition of a general framework for the specification and the evaluation of network security mechanisms/services against network security goals. In our approach we have defined a language that allows the expression of the network security objectives, the network security services and the network security mechanisms with their configuration. Moreover, it brings the ability to specify the network topology because the efficiency of the security mechanisms depends on. It also includes a formal evaluation process.

This article only presents our specification language, its expressiveness and our evaluation tool. The formal definition of the language and the formal evaluation process is presented by Laborde et al[15]. Section 2 exposes our way of defining the network security objectives. In section 3, we define our network model and our specification language and briefly comment the evaluation process. In section 4, we present our tool that automates the evaluation task by a simplistic example. Finally, in section 5, we conclude and introduce our plans for future works.

2. DEFINITION OF NETWORK SECURITY OBJECTIVES

Traditionally, the network security officer addresses security problems using an emperical approach where each problem is considered one after the other. Such a point of view does not permit to determine correct security objectives because they are not integrated in a global security management process. Management models, like the TMN[16] one, show clearly that networks provide services to applications. So, the requirements of the applications constitute the network objectives. Thus, in this section, we formalize this dependency in the security context.

2.1 The relationship between an application security policy and a network security policy

When a user accesses a service, a set of data flow is exchanged between the device from which the user launches the service and the devices supporting the service execution (fig. 1). So, a relation between a network security policy and an application security policy can be distinguished. For example, if the application security policy states that user "u_1" can read object "o_1"- noted (u_1, o_1, +read), then it implies that a corresponding data flow *flow(o_1,+read)* between the device of user "u_1" and the device of "o_1" can exists on the network. Consequently, the associated network security

policy must allows the data flows *flow(o_1, +read)* between these two devices – noted (device(u_1) ↔ device(o_1), +flow(o_1, read)). Conversely, if the application security policy states that user u_2 cannot read object o_2 noted (u_2, o_2, - read), there no flow *flow(o_2,read)* between the devices of u_2 and o_2. Therefore, the network security policy must forbid *flow(o_2, read)* between the devices of u_2 and o_2, i.e., (device(u_2) ↔ device(o_2), - flow(o_2, read)). We thus obtain the derivation relation noted "$\Rightarrow^d$" as $\forall u \in$ USERS, $\forall$ o $\in$ OBJECTS, $\forall$ a $\in$ ACTIONS, (u, o, ±a) $\Rightarrow^d$ (device(u) ↔ device(o), ± flow(o, a)).

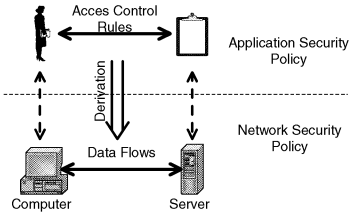

Figure 1. Security policy derivation

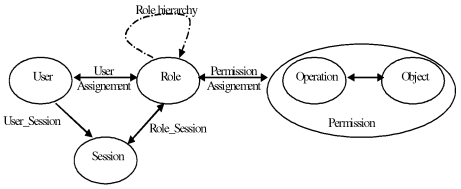

Figure 2. The NIST RBAC Model

2.2 The NIST RBAC Model

Access control is the process of mediating every request to resources and data maintained by a system and determining whether the request should be granted or denied. Access control models[1] provide a formal representation of the access control security policy and its working. The formalization allows the proof of properties[1,17-20] on the security provided by the access control system being designed.

Among the access control models, we have chosen the NIST RBAC model[17] because it simplifies the management tasks. Actually, the role concept allows aggregating the users' permissions and then it facilitates the users' rights modifications made by an administrator. Moreover, the hierarchies between roles represent a good tool for modeling an organization according to different points of view.

The NIST group proposes the standardization of the RBAC model. It is made up of two sub-models: the core model and the hierarchical model (fig.2).

The core model includes five sets of basic data elements:

- A *user* is an active entity, i.e., human or intelligent agent.
- A *role* is a job function within the context of an organization with some associated semantic regarding the authority and responsibility on the user assigned to the role. We can notice that the definition is very vague.
- A *permission* is an approval to perform an operation on one or more protected objects.
- An *operation* is an executable image of a program, which upon invocation executes some function on behalf of the user.
- An *object* is an entity that contains or receives information.

Finally, a set of roles is assigned to a user, and a set of permissions is assigned to a role. A session is a mapping of one user to a set of authorized roles.

The hierarchical model adds relations for supporting role hierarchies. There exist different approaches for constructing a role hierarchy: based on privileges[20] or based on users' job functions[21,22].

2.3 Towards an "RBAC network security policy"

Users are considered in an RBAC system by their assigned role. Consequently, the derivation relation becomes: $\forall$ r $\in$ ROLES, $\forall o_i \in$ OBJECTS, $\forall$ $op_j \in$ OPERATIONS, $\forall u \in$ USERS, $\forall u' \in$ USERS $\bullet$ $(r, \{(op_j, o_i)\}) \wedge$ Assigned(u,r) $\wedge \neg$Assigned(u',r) $\Rightarrow^d$ (device(u) $\leftrightarrow$ device(o_i), +flow(o_i, op_j)) $\wedge$ (device(u') $\leftrightarrow$ device(o_i), - flow(o_i, op_j)).

Thereafter, we consider that there is *no hierarchy* and that *roles have disjoint privileges* (if this is not the case then we may create a partition of this set): such a constraint will help us to group data flows based on the permissions assigned to one role and then identifying them by the role. Afterward, we note by the name of the role the set of flows corresponding to the permissions assigned to the role.

According to these definitions, we present our language which is able to express the network security objectives, i.e. the RBAC information, the network security mechanisms and the network topology.

3. NETWORK ARCHITECTURE MODEL AND NETWORK SECURITY SPECIFICATION

Each communication generates data flows between a source and a destination system. Our approach consider that the applicable treatments on data flow can be brought together into four basic functionalities. Devices are modeled while interconnecting these basic functionalities:

- Mechanisms that *consume/produce* data flows such as the end-systems,
- Mechanisms that *propagate* data flows such as physical supports and associated devices,
- Mechanisms that *transform* data flows into another one such as the security protocols,
- Mechanisms that *filter* data flows such as the firewall ones.

In our model (fig. 3), there are a set of active entities and a set of passive entities, and a set of functionalities (end-flow, channel, transform and filter) which act on information flows. An active entity corresponds to a user in the RBAC model, and a passive entity is a set of objects in the RBAC model.

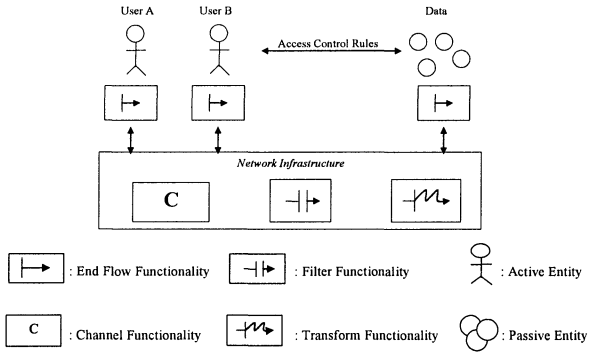

Figure 3. The network security and topology model

3.1 Definition of the functionalities

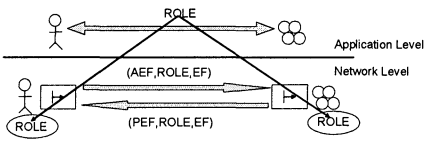

Figure 4. End-flow functionality

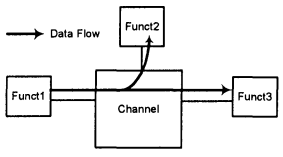

Figure 5. Channel functionality

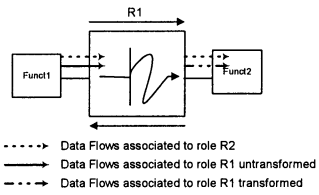

Figure 6. Transform functionality

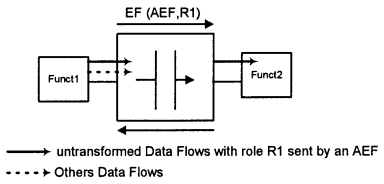

Figure 7. Filter functionality

We have modeled data flows and all basic functionalities using Colored Petri Nets. The formal definition is given in Laborde et al[15]. We just present here a non formal definition of each functionality:

The end-flow functionality.

An end-flow (EF) is a functionality that is specific to end-systems, i.e., data and application servers as well as the workstations. It constitutes the link between the application level security model, i.e. RBAC, and our network model. Hence, we consider two types of end-flow functionalities:

- *Active End Flow functionality* (AEF): An EF is said active if any active entity is connected to this EF.
- *Passive End Flow functionality* (PEF): An EF is said passive if any passive entity is connected to this EF.

We append a list of roles to each EF for indicating the flows that the EF can produce. The list corresponds to the set of roles assigned to the user representing the connected active entity for an AEF. In the case of a PEF, it is the set of roles assigned to the permissions that concern an object of the connected passive entity. When the users launch their authorized services, it implies a communication between all the AEF and the PEF with the same role (fig. 4). The flow produced by an AEF (resp. PEF) with role R is noted (AEF,R,EF) (resp. (PEF,R,EF)). It allows the expression of the network security objectives.

The channel functionality.

The channel functionality models the physical network. It receives the flow from one of the connected functionalities and retransmits it to all the others connected functionalities (fig. 5).

The transform functionality.

The transform functionality receives a data flow (ex: (AEF,R,EF)) from one of its two interfaces, according to transformation rules represented by a list of roles which identifies the data flows that must be transformed, and sends to the other interface the same data flow or the data flow transformed

represented by the parameter TR (ex: (AEF,R,TR)) (fig. 6). This new flow has the confidentiality, integrity and authenticity properties.

The filter functionality.

The filter functionality (fig. 7) stops or forwards a data flow. We find this functionality in firewalls, Application Level Gateways or filtering routers. We restrict it to only connect two functionalities. The filtering rules explicitly express the permitted flows between its two interfaces; if they are preceded by "EF" they come untransformed from an end-flow functionality, else if they are preceded by "TR" then they have been modified by a transform functionality.

3.2 Security analysis

The CPN model associated to each specification produces a reachability graph. It is analyzed with the set of security properties described here after. The formal properties definitions, the analysis process, its applicability in complex studies are given in Laborde et al[15].

Property of confidentiality.

Basically, the property of confidentiality limits protects the data from unauthorized disclosure. Thus, in our model, it prohibits an end-flow functionality from receiving at any time a untransformed data flow with any unassigned role.

Property of integrity.

Classically, the property of integrity prohibits non granted entities from any creation, modification or destruction of objects. Then, in our model, this property lay down that an end-flow functionality can only generates data flows with its assigned roles.

Property of availability.

This property stipulates that all the granted services must be available to all the authorized entities. In the network environment, the data flows corresponding to this must be able to travel between both devices. Consequently, its translation in our model is all active (resp. passive) end-flow functionalities must be able to consume all the data flows with an assigned role sent by every passive (resp. active) end-flow functionalities.

As we intend to address devices configurations, we complete these classical security properties with new ones:

Property of partitioning.

It is used to limit to the propagation of data flows. It declares that a data flow can only pass a filter functionality if it is situated between the data flow source and a possible correct destination.

Non productive filtering rule.

It is used to eliminate unnecessary filtering rules. Let f, a filter functionality connected to the functionalities fct_1 and fct_2. We say that the filtering rule which let pass a data flow from fct_1 to fct_2 is non productive if this flow never try to pass through the filter functionality.

Non productive transform rule.

This one is used to eliminate unnecessary transform rules. A transform rule tf that transforms the data flows with the role "r" from fct_1 to fct_2 is non productive, if any flow with the role "r" pass through the transform functionality in the direction fct_1 to fct_2 at any time.

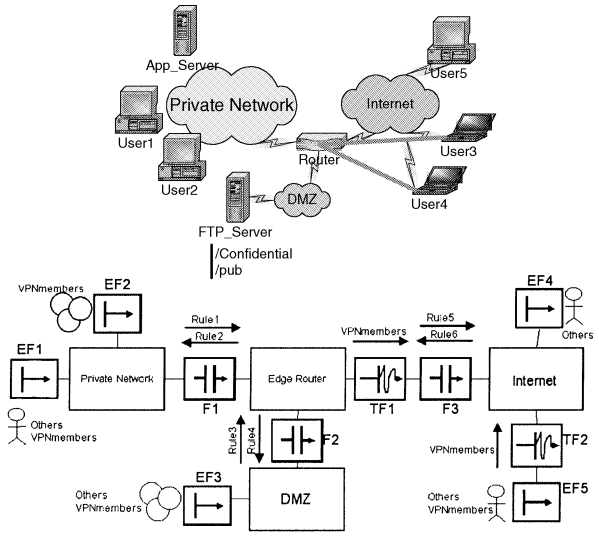

Figure 8. Architecture and graphical specification of our VPN example

4. A NETWORK SECURITY POLICY EVALUATION EXAMPLE

Like in traditional enterprise network, this example considers an edge router interconnecting a private network and a DMZ. An "App_Server" server and an FTP server are respectively installed in the private network and in the DMZ (fig. 8). The application level security policy is a RBAC one, without hierarchy, where two user groups "*VPNmembers*" and "*Others*"

are defined. This organization is only based on the granted privileges. The "App_Server" server is dedicated only to the services usable by the *VPNmembers* group. The FTP_Server has two directories: /confidential and /pub. The directory "confidential" contains data only accessible to the *VPNmembers* users group. Data of the "pub" directory is accessible to everyone. $User_1$, $User_2$, $User_3$ and $User_4$ belong to *VPNmembers* and *Others* groups. $User_5$ is only member of the *Others* group.
The application level security policy can be expressed as:

Permissions(VPNmembers) = {(+all_access, FTP_Server/confidential),
(+all_access, App_Server)}
Permissions(Others) = {(+all_access, FTP_Server/pub)}

Figure 8 shows the network topology specification and the network level security policy implemented in our language. The filtering rules associated with the filter functionalities of our example are:

- Rule1 = EF (AEF, Others) (AEF, VPNmembers)
- Rule2 = EF (PEF, Others)
- Rule3 = EF (PEF, Others), (PEF, VPNmembers), (AEF, VPNmembers)
- Rule4 = EF (AEF, Others), (AEF, VPNmembers)
- Rule5 = EF (PEF, Others) (AEF, Others)
  TR (PEF, VPNmembers)
- Rule6 = EF (AEF, Others)
  TR (AEF, VPNmembers)

We have developed using Java programming language a tool that automates the evaluation task. It takes as an input a specification file. First, it analyzes the syntax. If the syntax is correct, it generates the equivalent CPN and checks all the properties. Finally, it produces as a result a file (fig. 9) that deals with if the properties are satisfied or not. If a property is not satisfied, the reason is explained.

In our example, the tool indicates (fig. 9) that the property of confidentiality is satisfied and there is no non productive transform rule. Nevertheless, the availability is not satisfied because ef_2 cannot receive any flow with the role VPNmembers from ef_5, ef_1 cannot receive any flow with the role VPNmembers from ef_3 and ef_5 cannot receive any flow with the role VPNmembers from ef_2. The partitioning properties is not satisfied on account of the rule EF (AEF,Others) from tf_1 to Internet in the filter functionality f_3. And finally, the filtering rule EF (AEF, VPNmembers) from dmz to edge_router in the filter functionality f_2 is non productive. To resume, this specification is not secure.

```
Property of Confidentiality :
-----------------------------------
ef5 : OK
ef4 : OK
ef1 : OK
=> The property of confidentiality is satisfied

Property of Availability :
--------------------------------
ef5 :
   no flow with the role vpn-members from ef2
ef4 : OK
ef1 :
   no flow with the role vpn-members from ef3
ef3 : OK
ef2 :
   no flow with the role vpn-members from ef5
=> The property of availability is not satisfied

Partitioning Property :
----------------------------
f3 :
   Rule 1 -> 2 :
     [ EF (AEF ,others) ]
   Rule 2 -> 1 : OK
f2 :
   Rule 1 -> 2 : OK
   Rule 2 -> 1 : OK
f1 :
   Rule 1 -> 2 : OK
   Rule 2 -> 1 : OK
=> There is one or more partitioning problem

Non Productive Transform Rules :
-------------------------------------------
tf2 :
   rules 1 -> 2 : OK
   rules 2 -> 1 : OK

tf1 :
   rules 1 -> 2 : OK
   rules 2 -> 1 : OK
=> There is no non productive rule

Non Productive Filtering Rules :
-----------------------------------------
f3
   rules 1 -> 2 :  OK
   rules 2 -> 1 :  OK
f2
   rules 1 -> 2 :
      [ EF (AEF, vpn-members) ],
   rules 2 -> 1 :  OK
f1
   rules 1 -> 2 :  OK
   rules 2 -> 1 :  OK
=> There is one or more non productive rule
```

Figure 9. Evaluation result file

5. CONCLUSION

The work presented here combine different levels of policy abstraction and security analysis coming from new management approaches and the formal modeling and evaluation techniques. The quiet simple language that we have proposed allows to formally evaluate the network security policy, while using the underlying CPN powerful.

At present, we are testing our approach through different case studies to enhance our method. We are focussing on validating the real configurations on devices. As our tool is independent from the security technologies implemented on the devices, it confines itself to only validate security mechanisms constraints. The next usefull step is to bridge this gap thanks to the Common Information Model[23] defined by the DMTF task force to harmonize the management systems. Hence, we could interconnect our work with management platforms.

REFERENCES

1. 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.
2. Guttman J., "Filtering postures : Local enforcement for global policies", IEEE Symposium on Security and Privacy, Oakland CA, USA, 1997.
3. Ehab Al-Shaer and Hazem Hamed, "Discovery of Policy Anomalies in Distributed Firewalls", in IEEE INFOCOMM'04, March 2004.
4. Y. Bartal., A. Mayer, K. Nissim and A. Wool. "Firmato: A Novel Firewall Management Toolkit." proceedings of 1999 IEEE Symposiumon Security and Privacy, May 1999.
5. Guttman J., Herzog A., Thayer F., "Authentication and confidentiality via IPsec", 6th European Symposium in Computer Security ESORICS, Toulouse, France, 2000.
6. Z. Fu, F. Wu, H. Huang, K. Loh, F. Gong, I. Baldine and C. Xu."IPSec/VPN Security Policy: Correctness, Conflict Detection and Res-olution." Proceedings of Policy'2001 Workshop, January 2001.
7. Yavatkar R., Pendarakis D., Guerin R., "A Framework for Policy-based Admission Control", RFC 2753, January 2000.
8. Jennings N.R., Bussmann S., "Agent based Control Systems, why are they suited to engineering complex systems? IEEE Control Systems Magazine, vol 23, No 3, June 2003.
9. IBM Corporation, An Architectural Blueprint for Autonomic Computing, IBM white papers, April 2003.
10. http://www.solsoft.com
11. Hinrichs S., "Policy Based Management : bridging the gap", in 15th Annual Computer Security Applications Conference (ACSAC 99), December 1999.
12. Westerinen A., Schnizlein J., Strassner J., Scherling M., Quinn B., Herzog S., Huynh A., Carlson M., Perry J., Waldbusser S., "Terminology for Policy-Based Management", RFC 3198, November 2001.
13. Moore B., Ellesson E., Strassner J., Westerinen A., "Policy Core Information Model -- Version 1 Specification", RFC 3060, February 2001.
14. Arosha K Bandara, Emil C Lupu, Jonathan Moffet, Alessandra Russo,. "A Goal-based Approach to Policy Refinement", in: Policy 2004, June 2004.
15. Laborde R., Nasser B., Grasset F., Barrère F., Benzékri A., "A formal approach for the evaluation of network security mechanisms based on RBAC policies", In WISP'04, Electronic Notes in Theoretical Computer Science, Elsevier, to appear.
16. "Principles for a Telecommunications Management Network", ITU-T, M3010, May 1996.
17. " Role-Based Access Control", ANSI/INCITS 359-2004, February 2004.
18. R. Peri, "Specification and verification of security policies", PhD Dissertation, University of Virginia, January 1996.
19. Wijesekera D., Jajodia S., "A propositional policy algebra for access control", ACM Transactions on Information and System Security (TISSEC), vol 6,2003.
20. Nyanchama M., Osborn S., "The role graph model and conflict of interest", ACM Transactions on Information and System Security (TISSEC), vol. 2, 1999.
21. Moffett J. D., "Control Principle and Role Hierarchies", 3rd ACM Workshop on Role Based Access Control, Fairfax, VA, 1998.
22. Crook R., Ince D., Nuseibeh B, "Modeling Access Policies using Roles in Requirements Engineering", Information and Software Technology, 2003, Elsevier
23. http://www.dmtf.org/standards/cim