

Improving Packet Filters Management through Automatic and Dynamic Schemes

Olivier Paul[†], Maryline Laurent
*ENST de Bretagne, Département RSM,
2 rue de la châtaigneraie,
35510 Cesson-Sévigné, FRANCE.
{Olivier.Paul|Maryline.Laurent}@enst-bretagne.fr*

Key words: Access Control, Management, Packet Filters, Security.

Abstract: The development of complex access control architectures raises the problem of their management. In this article, we describe an architecture providing packet filters configuration in Internet based networks. The performance of the access control process heavily depends on the number of rules used to define the access control service. Therefore an efficient access control architecture relies on a clever access control rule configuration. The current approaches to this problem are based on a static description of the network which can force security officers to choose between efficiency and manageability. Our distribution architecture doesn't rely on a model of the network and thus eliminates this limitation while proposing three optimisations in order to provide the access control processes with optimal configurations.

1. INTRODUCTION

With the deployment of the internet, security in general and access control in particular are becoming major concerns in the design of communication networks. In particular, improvements in the networking technology in term of throughput has raised the problem of the performance of access control mechanisms that have been developed in the past. As a

[†] This work is funded by DRET.

consequence much interest has been paid in the recent past to the development of new algorithms to improve packet filters ([12],[13]). As stated in [11], the packet classification based on d header fields among N rules is a problem where, on one hand, the best algorithm in terms of time complexity has a $O(\log(N))$ complexity but requires a $O(N^d)$ working space. On the other hand, the best algorithm in terms of space complexity only needs a $O(N)$ space but has a $O(\log^{d-1} N)$ time complexity. Even when trade-offs are used between both of these algorithms, the number of rules (N) which are used to define the access control policy has a great impact either on the time complexity of the system, either on its space requirements.

In this article we describe an architecture configuring packet filters automatically with a small subset of access control rules in order to increase the filter performances. Another key point of our architecture is its ability to work without relying on a static description of the network unlike existing solutions. Section 2 gives details about the current solutions for automatic access control devices management. We then describe in section 3 the components of our access control management architecture. Section 4 presents the results of our architecture testing using the *ns* simulator. Finally section 5 compares the benefits and drawbacks of such an architecture over the existing solutions and outlines future works.

2. CURRENT APPROACHES

The current approaches can be classified into three classes according to the place where the access control rule distribution process takes place and the level of optimisation included into the distribution process.

Centralised and automatic with weak optimisation

[6] is the first article introducing the concept of an automated distribution process. This distribution process is based on two parameters which are the access control policy and a model of the network. However the model of the network allows the network to include cyclic paths. These cyclic paths generate non optimal configurations since access control rules can be distributed to devices that will never be reached by the communications described by the access control rules since they are not topologically on the path used by the communication.

Centralised and automatic with optimisation

In order to cope with the inefficiency of a cyclic model of the network, [1], [7] and [10] rely on a tree description of the network. They also use a description of the device access control capabilities in order to prevent the

rules to be assigned to a device where the rule can not be executed. However this approach forces the security officer to reconfigure the network model after each topology change in order to reflect the state of the network. Despite this drawback, this approach is the most popular today.

Distributed and automatic with optimisation

[8] suggests that access control information could be distributed across the network in a distributed fashion following the routing information distribution method. As an example, the authors propose a Packet Filter Information Protocol (PFIP) to manage filtering routers located within a network. This approach is interesting since it provides a way to configure a distributed architecture through a distributed protocol without relying on a description of the network. However the method proposed in this paper aims to optimise the use of the network and does not consider the optimisation of the access control process itself.

To summarise this section, we can state that the current proposals force security officers to make a decision between the optimisation of the access control process and the usability of their access control management architecture. This is the reason why we focus in this article on the optimisation of the distribution and configuration process and show how to improve and implement the optimisations proposed by [4] without relying on a description of the network. In the next sections we assume that a generic language has been defined to define access control policy. This language allows the security officer to define an access control policy for a whole network through a set of rules.

3. DISTRIBUTION ARCHITECTURE

Since our optimisations rely on the content of the access control rules, we need to specify how access control rules can be defined. Each rule includes a set of conditions and one action. Each condition consists of one access control object, one operator and one access control value or a range of access control values. The action usually specifies if the communication has to be permitted or denied.

3.1 Overview

Our architecture is based on modified management agents. These agents are located on packet filters and are assumed to configure each packet filter with an optimal configuration. An optimal configuration means that the

packet filter is configured with the smallest subset of access control rules that can be used to provide the global access control policy. In order to achieve this optimal configuration, we define a set of assertions that insure this property.

- (Assertion 1) A rule must not be assigned to an agent where this rule cannot be executed. A rule cannot be executed when an access control object required by the rule is not supported by the packet filter.
- (Assertion 2) A rule must not be assigned to an agent where this rule will never be executed. This case occurs when the rule is not located on the path between the source and the destination described by the rule.

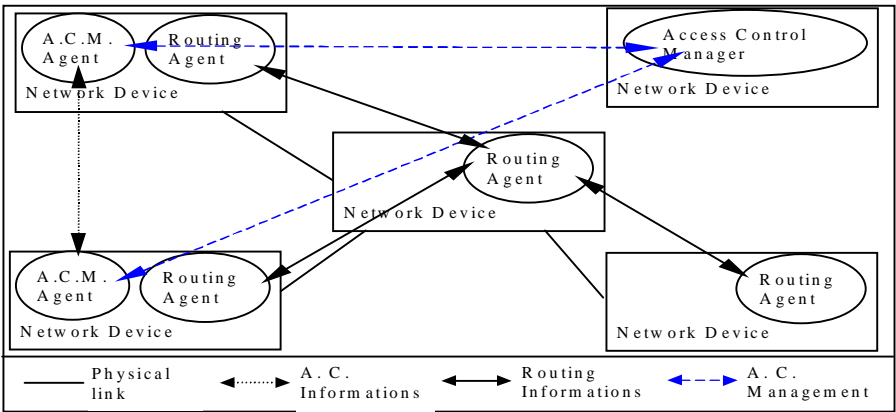


Figure 1. Interactions between network elements.

Providing more information about our security policy allows us to optimise our distribution process. This optimisation is based on the fact that the type of our access control policy is «What is not explicitly permitted is prohibited». With this kind of access control policy, a «Deny rule» always describes a subset of a «Permit rule». As a consequence a single «Deny rule» can block a communication. This type of access control policy is the most common but the opposite type (i.e. «What is not explicitly prohibited is allowed») would result in a similar optimisation. In order to insure a better distribution of the access control rules and a smaller subset of rules to be assigned to each packet filter, each rule has to be assigned to the packet filter which is the closest to the source or the destination described by the rule. This property results from the general tree structure that can be found in networks. Our last optimisation applies to «Deny rules». The distribution of these rules must follow assertion 3:

- (Assertion 3) If a «Deny rule» can be assigned to several cascading packet filters, the rule has to be assigned to the packet filter which is the closest to the end devices.

The translation of these assertions into a real algorithm can be done by using parameters such as the network topology which may change with routing information, the packet filter access control capabilities, the position of the packet filter in the network topology and the configuration of other packet filters. These parameters imply that each agent has to interact with other network elements. Some of these interactions are located inside the packet filter device. Some others described in figure 1 are external.

The next section describes the internal structure of our access control management agents.

3.2 Functional Architecture

As described in figure 2, our architecture is based on several modules interacting with each other.

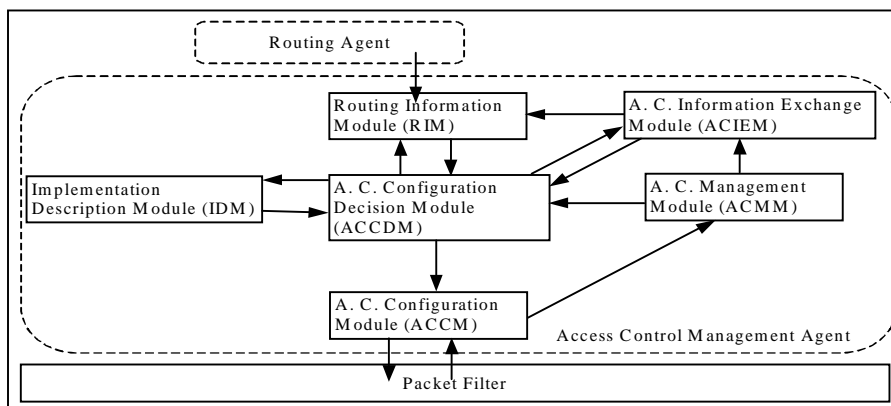


Figure 2. Agents internal structure

- The Access Control Configuration Module (ACCM) provides an interface between the Access Control Agent and the packet filter. The goal of this module is to translate the rules received from the Access Control Configuration Decision Module (ACCDM) into commands according to the packet filter syntax. When the packet filter has been configured, the module receives the results from the execution of the access control rules and sends these results to the Access Control Management Module (ACMM).
- The Implementation Description Module (IDM)'s goal is to keep information about the access control capabilities of the packet filter where it is located. Consequently, the IDM is able to tell the ACCDM if an access control rule can be implemented by the packet filter. The IDM

also informs the ACCDM of changes in the access control implementation configuration.

- The Access Control Management Module (ACMM). The aim of this module is to allow the Access Control Management Agent to communicate with the Access Control Manager (ACM). In order to complete this task this module manages a Management Information Base (MIB) dedicated to access control management. This MIB is initialised by the ACM with all the access control rules at the beginning of the access control configuration process. The MIB is then updated by the ACM when changes are made to the global access control policy. The results from the access control rules execution by the packet filter are provided to the ACMM by the ACCM. These results are stored in a MIB table and retrieved periodically by the ACM.
- The Access Control Information Exchange Module (ACIEM) 's goal is to answer the following question to the ACCDM: “Is there another device implementing the rule r more efficiently?”. In order to answer this question, the ACIEM interacts with its ACIEM neighbours through the Access Control Information Exchange Protocol (ACIEP). This protocol allows the election of an agent which is chosen to implement a rule. For each rule, the ACIEM compares the local information with the information provided by its neighbours and decides if its agent is located at an optimal position to implement the rule. An optimal position means that the agent is closer to the source or closer to the destination than any other agents. The ACIEM also notifies the ACCDM in the case of topology changes or when modifications in the configuration of packet filters occur.
- The Routing Information Module (RIM) is used by the ACCDM to know if a communication described by the rule r goes through the device where the agent is located. In order to provide this answer, the RIM uses the fact that a communication goes through a routing device if this device is located on the shortest path between the source and the destination. This information can be found by comparing the routing information provided by neighbour nodes to the information provided by the routing table. Moreover, the RIM has to provide the ACIEM with the distance between the device hosting the RIM and another device. The RIM also alerts the ACCDM when it detects some topology changes.
- The Access Control Configuration Decision Module (ACCDM) is the main module of our architecture. The ACCDM is alerted by the ACMM when new rules are sent to the agent by the ACM. The ACCDM checks the three assertions defined in section 3.1 for each new rule by interacting with the IDM, ACIEM and RIM. Rules complying with our three assertions are sent to the ACCM for implementation.

4. SIMULATION RESULTS

The *ns* simulator [5] is a discrete event simulator targeted at networking research. *ns* provides substantial support for simulation of TCP, routing, and multicast protocols. Using the *ns* simulator, we made an implementation of the architecture presented in the previous section. Interested readers can find a package including this implementation and several test suites on our web page [9]. In the next section we define a “typical” access control policy and show the results from the distribution of this policy on a meshed network. These results allow us to prove the efficiency of our distribution method and to analyse the improvements provided by each assertion.

Access control policy modelling

In order to have interesting simulation results we used the structure of our simulation software to represent real access control rules. We took examples from [3] and [2] which present typical access control policies for various internet services and then added some changes in order to reflect what can be found in real configurations. The resulting policy is made of 80 rules. 10 of these rules are dedicated to prevent attacks that are not linked to a specific address (typically DoS attacks) whereas the 70 other rules are dedicated to control the access to internet services (mail, dns, www, ftp, telnet ...) and are always linked to specific addresses or range of addresses.

In order to model these rules we classify access control rules according to two parameters. The first one is the action specified by the rule. This action can permit or deny the communication. As a result we have two classes of rules called ALLOW and DENY. The second one is the addressing information described by the rule. Rules providing addressing information (i.e. a source and a destination descriptors) are called specified rules whereas rules not providing addressing information are called unspecified rules.

Table 1. Rules classification

Rule types	ALLOW	DENY
Unspecified	5	5
Specified	60	10

Meshed network example

Our example is based on a meshed network representing a company’s private network. The network is made of 39 nodes. 27 of these nodes are “terminal” and represent sub-networks. The other 12 non-terminal nodes represent the core network. These two classes of nodes have various access control requirements. Terminal nodes are supposed to provide the access control service for the whole network they are representing whereas non-

terminal nodes protect the core network against denial of service attacks. As a result the nodes belonging to the core network only implement the unspecified rules whereas the terminal nodes are requested to implement the global access control policy.

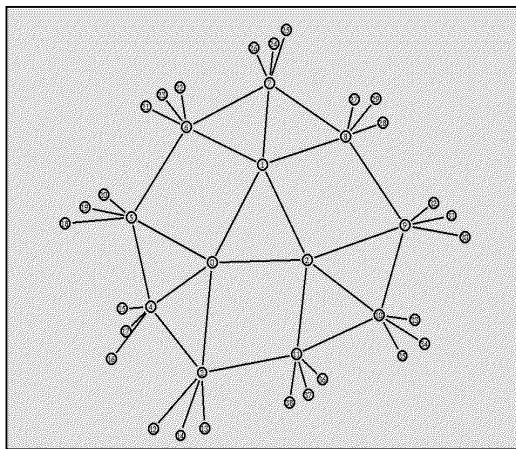


Figure 3. Network topology.

Specified rules are specific to each network to protect. As a result, the instantiation of the access control policy on our network topology includes 27 sets of 70 specified access control rules. The unspecified rules are common to all packet filters and are represented once in the global access control policy. As a consequence, the resulting policy includes 1900 rules.

Table 2. Distribution methods efficiency

Distribution method	Number of rules	avg. num. rules/node	Loss (%)
Brute	74000	1900	0
Based on capabilities	51420	1318	30.6
Based on routing information	9691	248	86.9
Based on routing information and action type	8679	222	88.3
Complete optimisation	3842	99	94.8

The results from our simulations are presented in table 2. These results show the relative efficiency of the enforcement of our assertions. We don't provide results for the optimisation only based on the action type since this optimisation cannot be implemented without the optimisation based on routing information.

The improvements brought by our optimisation schemes are significant since the average number of rules per node is divided by 20 with the

complete optimisation. More generally, additional simulations suggest that the complexity of the access control policy implemented by each packet filter is reduced from $O(n \cdot m)$ with a brute configuration to $O(n)$ where n is the number of rules to protect each terminal network and m the number of networks to protect. However our simulations show that our architecture fails to configure the packet filters with n rules. This problem can be explained by a simple reason: the distribution of each specified “permit” rule generates the configuration of two packet filters, the first one is the packet filter which is the closest to the source and the second one is the packet filter which is the closest to the destination. However this problem could be solved by using rules aggregation.

5. CONCLUSION

In this paper we presented an architecture designed to automatically distribute and configure an access control policy over a network including several packet filters. We showed that our architecture is able to configure these devices with relevant and efficient configurations.

In comparison with other approaches our architecture has several advantages.

- It allows the security officer to define an access control policy once and to have this access control policy distributed automatically on packet filters.
- Our architecture proposes three optimisations producing efficient access control configurations.
- The security officer doesn’t have to manage the complexity introduced by the network topology since the distribution of the access control rules automatically follows the network topology changes.
- The architecture is distributed providing a more scalable, efficient and reactive solution.
- The security officer can control the application of his access control policy since our architecture can give feedback providing the security officer with the rules that have been implemented and those that have been left unimplemented.

Our work could be usefully continued in several directions. The first one is to improve the security of our architecture by extending the access control information exchange protocol to provide authentication, integrity and a protection against replay attacks. These services appear necessary to prevent attacks against our architecture. The last one would be to extend our current

architecture to allow other kinds of access control tools using rule based configurations (e.g. wrappers at the application level) to be managed.

6. REFERENCES

- [1] Firmato, A Novell Firewall Management Toolkit, Yair Bartal, Alain Mayer, Kobbi Nissim, Avishai Wool, IEEE Symposium on Security and Privacy, Oakland, May 1999.
- [2] Building Internet Firewalls, B. Chapman, E. Zwicky, O'Reilly & Associates, 1995.
- [3] Firewalls and internet security, repelling the wily hacker, B. Cheswick, S. Bellovin, Addison-Wesley publishing company, 1994.
- [4] Integrated Management of Network and Host Based Security Mechanisms, R. Falk, M. Trommer, 3rd ACISP, Brisbane, Australia, July 1998.
- [5] ns Notes and Documents, Kevin Fall, Kannan Varadhan, September 1999.
- [6] Filtering Postures: Local Enforcement for Global Policies, Joshua D. Guttman, IEEE Symposium on Security and Privacy, Oakland, May 1997.
- [7] Policy-Based Management: Bridging the Gap, Susan Hinrichs, 15th Annual Computer Security Applications Conference, Phoenix, December 1999.
- [8] Management of Network Security Application, P. Hyland, R. Sandhu, 21st National Information Systems Security Conference, October 1998.
- [9] <http://www.rennes.enst-bretagne.fr/~paul/acm.zip>
- [10] An Asynchronous and Distributed Access Control architecture for ATM networks, Olivier Paul, Maryline Laurent, Sylvain Gombault, ACSAC'99, Phoenix, December 1999.
- [11] A Novel Hardware Cache Architecture to support layer-four Packet Classification at Memory Access Speeds, J. Xu, M. Singhal, J. Degroat, Technical report, February 1999.
- [12] Packet Classification using Tuple Space Search, V. Srinivasan, S. Suri, G. Varghese, ACM Sigcomm'99, September 1999.
- [13] High-Speed Policy based Packet Forwarding Using Efficient Multi-dimensional Range Matching, T.V. Lakshman, D. Stiliadis, ACM Sigcomm'98, September 1998.

7. BIOGRAPHIES

Olivier Paul received a DEA in computer science from the university of Rennes in 1996. He is currently working toward his Ph.D. within the Network and Multimedia Services department of ENST Bretagne. His research interests include high speed networks, access control mechanisms and architectures as well as security management.

Maryline Laurent obtained her PhD in 1997. Since then she has been working in the Networks and Multimedia Services Department of ENST Bretagne as an associate professor and in project ARMOR of INRIA as an expert scientist. Her main research area is networks security, mainly in the ATM and IP fields. She contributed to the ATM Forum and participated in the European project SCAN (Secure Communications in ATM networks).