



Improving Network Access Control Integrity Through Redundant Mechanisms

Olivier Paul

► To cite this version:

Olivier Paul. Improving Network Access Control Integrity Through Redundant Mechanisms. IFIP International Conference on Information Security IFIP/SEC'2000, Aug 2000, Pékin, China. <hal-03643663>

HAL Id: hal-03643663

<https://hal.science/hal-03643663v1>

Submitted on 16 Apr 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

Improving Network Access Control Integrity Through Redundant Mechanisms

Olivier Paul[†]

ENST de Bretagne

Olivier.Paul@enst-bretagne.fr

Abstract

Distributed access control schemes usually use automatic access control configuration schemes. These methods rely on three criteria to improve the positioning of access control rules. In this paper we show that the alteration of one of these criteria can bring some redundancy in the access control configuration. This redundancy can be used to improve the trust a security officer can have in the access control process integrity. We demonstrate that the overhead generated by this redundancy is generally negligible when the underlying network is sufficiently big.

Key-Words

Access Control, Integrity, Management, Security.

1. Introduction

With the deployment of the internet, security in general and access control in particular are becoming a major problem in the design of communication networks. As a consequence, much attention has been paid to develop mechanisms to provide the access control service for every kind of networks. These developments have been mainly conducted by security working groups within companies and academic organizations and gave birth to the today's most common network access control tool, called firewall.

Today, a new kind of firewall is appearing ([12], [11], [10]). This technology relies on a distribution of the access control process on several access control devices. This distribution has several advantages (performance, security and quality of service preservation). However some of these devices may be end systems and can be logically or physically modified by users. As a result, the integrity of the access control process is an important issue. How can the security officer be sure that the access control

process is implemented correctly ? In this paper we show how the integrity of the access control process can be insured by adding some redundancy on the configuration of access control devices. This paper is organized in four parts. We first describe in section 2 some of the distributed access control proposals. We then describe a scheme that gives the security officer some insurance about the access control process integrity.

We provide in section 4 computational and simulation results showing that the overhead generated by our scheme can be very low with networks including a large number of access control devices which is usually the case in distributed access control architecture.

Finally section 5 compares the benefits and drawbacks of such optimization and talks about future works.

2. Distributed Access Control

The main difference between distributed access control mechanisms and traditional access control mechanisms used by firewalls is the place where the access control process takes place. By opposition to traditional access control devices who provide the access control service in the border between a public unsafe network and a private secured network, distributed access control schemes propose to provide the access control service through a set of access control tools distributed across the network. For example [12] suggests to place the access control in the network internal devices by using the access control capabilities of these devices at the LLC, network and transport levels. [11] suggests to locate the access control process in end devices by taking advantage of operating systems protocol stacks access control capabilities. This proposal also show how to provide some access control at the application level through relevant software configuration. [10] proposes to provide the access control on internal devices and on end devices through software agents. These agents are requested to watch and control ongoing communications in a non blocking way.

In order to provide each access controller with a relevant access control policy, the access control

[†] This work is funded by DRET.

policy is generally globally defined by the security officer and then distributed to access control devices. However a brute distribution of the access control policy can result in a very inefficient access control process. As a result, access control distribution methods have been defined to configure each access control device with the smallest subset of access control rules allowing the access control policy to be enforced. These access control distribution methods ([3], [4], [5], [6], [7], [10]) are usually based on three parameters:

- The network topology.
- The rules content.
- The devices access control capabilities.

We show in the next section how these distribution methods can be lightly modified to give the security officer some insurance about the integrity of the access control process.

3. Proposed Solution

Figure 1 provides an example of an access control policy allowing the workstation with address 192.165.203.5 to communicate with the WWW server located on station 121.6.7.3. The policy is expressed by a set of rules, each rule describing a part of the communication.

```
ipfwadm -F -a accept -b -P tcp -S
192.165.203.5      1024:65535      -D
121.6.7.3 80
ipfwadm -F -a deny -b -P tcp -S
121.6.7.3 80 -k -D 192.165.203.5
1024:65535
ipfwadm -F -a accept -b -P tcp -S
121.6.7.3 80 -D 192.165.203.5
1024:65535
```

Figure 1. Access Control Policy

The current optimizations take advantage of the routing protocols to restrict the set of devices that have to be configured with an access control rule. Since the routing information defines the path between the source and the destination of the communication expressed by a rule, the distribution of the access control rule describing a communication can be restricted to the nodes located on this path. The routing information is usually static and provided by the security officer at configuration time.

The second optimization (later called rule type optimization) applies when the type of the access control policy is “What is not explicitly permitted is prohibited”. In this case, a “Deny rule” always describes a subset of a “Permit rule”. As a consequence, a single “Deny rule” can be used to

block a communication. This means that a single device on the path between the source and the destination described by a rule has to be configured with the “Deny rule”. On the other hand each “Permit rule” has to be assigned to each device on the paths between its source and destination.

Our integrity scheme is based on this last optimization. We claim that allowing some redundancy on “Deny rules” can provide some insurance about the integrity of the access control process. This insurance is particularly strong when a physically secured access control device enforces one of these rules.

This property can be explained by several reasons. Taking control of an end system can be quite easy when this system is not protected through tamper proof hardware. However taking control of a system that is not physically accessible is a much more difficult task. If we consider that users only have a physical access to their own personal computer we can claim that the access control processing that is not performed by their own computer can be classified as safer. Some access control management architectures assign “Deny rules” to end devices. As a consequence these rules can easily be bypassed by the computer owner. However other access control rules are much more difficult to bypass because these rules are generally attributed to other access control devices across the network. As a consequence configuring other access control devices with “Deny rules” is sufficient to block communications that could be generated by a user bypassing the access control process located on his computer.

Our approach is different from the classical centralized firewall approach since:

- The access control process remains distributed. As a consequence the performance speedup provided by the distribution is not avoided. Access control devices only enforce a part of the access control policy.
- As demonstrated in section 4, the overhead generated by the duplication of “Deny rules” is generally small.
- The “Deny rule” based optimization can be partially used by limiting the duplication of “Deny rules”.
- The security officer has the ability to make a tradeoff between performance and integrity by choosing the level of access control redundancy for “Deny rules”.

The overhead generated by our scheme is based on two parameters:

- The proportion of “Deny rule” in the access control policy.

- The size of the network since the size of the network will change the average length of the paths of the communications described by the access control rules.

From an implementation point of view, the changes introduced by our approach remain small since a single test on the number of devices configured with a rule has to be included in the existing optimization algorithms.

We show in the next section the impact of our proposal on the current distribution process performance.

4. Simulation Results

In order to evaluate the impact of the access control integrity on the current access control distribution performance, we simulate two distribution processes. The first one is based on a single optimization parameter (topology) whereas the second one is based on both optimization criteria (topology and type of the rule). We then compare both optimizations in order to evaluate the overhead generated by the redundancy. The overhead computed in this section is a worse case overhead that could be reduced by using a partial implementation of the type of rule optimization.

Our simulations are based on an implementation of algorithms implementing both criteria with the ns simulator [8]. The ns simulator is a discrete events simulator targeted at network research. The resulting implementation is made of about one thousand lines in O-tcl and one hundred lines in C++. A complete package including the implementation and several test suites is available at [9].

Access control policy modeling

In order to have interesting simulation results we used the structure of our simulation software to represent real access control rules. Since access control policies can greatly vary from one site to another, defining a typical access control policy is not an easy task. Moreover the language used to define the policy can bring a lot of changes in the way to express the access control policy. As a result we took examples from [2] and [1] which present typical access control policies for various internet services and we then added some changes in order to reflect what can be found in a real configuration.

In order to model these rules we classify access control rules according to two parameters:

- 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 addressing information described by the rule.
 - Rules providing addressing information (i.e. a source and a destination descriptors) are called specified rules.
 - Rules not providing addressing information are called unspecified rules.

Rule types	ALLOW	DENY
Unspecified	5	5
Specified	60	10

Table 1. policy1 description

To reflect real configurations, we consider two opposite kinds of policies. A first one called policy1, described in table 1 includes a small number of DENY rules. A second one called policy2 described in table 2 includes a large number of DENY rules.

Rule types	ALLOW	DENY
Unspecified	5	5
Specified	40	30

Table 2. policy2 description

Theoretical results

The network used in our simulation is well known and based on a tree of order 3 topology. Therefore, evaluating theoretically the impact of the type of rule criterion is possible if we suppose that each node is able to implement access control functions. We can then check if our simulations match our calculus in order to verify their accuracy. As stated in the previous section, the overhead generated by the use of the criterion depends on the size of the network and on the access control policy composition.

The average number of rules produced by the distribution of an access control policy by using the both optimizations can be expressed by the following function:

$$Nr0 = u \cdot n + avgl(n) \cdot sa + sd$$

Where u is the number of unspecified rules, sa the number of specified allow rules, sd the number of specified deny rules, and $avgl(n)$ is the average length between two leaves in a tree of order 3 with n nodes.

When a single optimization is used the average number of rules can be expressed by:

$$Nr1 = u \cdot n + avgl(n) \cdot (sa + sd)$$

As a consequence the gain involved by the type of rule criterion can be expressed by:

$$G = Nr1 - Nr0 = sd \cdot (avgl(n) - 1)$$

Simulation results

In order to evaluate the impact of our scheme on the distribution we make the network size vary from 4 to 121 (4, 13, 40 and 121) nodes and simulate the distribution for each policy. We assume that each node

is able to implement access control functions. For each simulation we compare the results based on our two criteria (topology and type of rule) with the results obtained with a single criterion (topology).

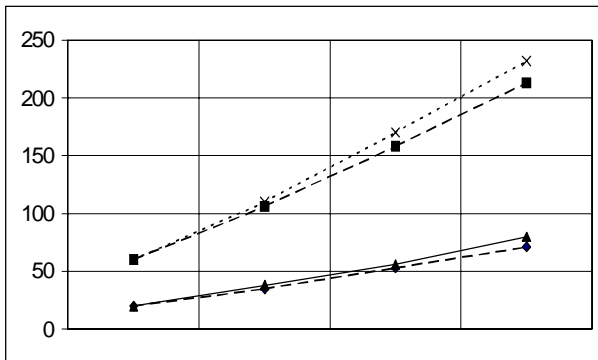


Figure 2. Criterion efficiency (in rules)

Figure 2 provides the gain in rules when two criteria are used. The results for policy1 are displayed by the plain line whereas the results for policy2 are displayed by the dotted line. The figure shows that the gain increases with the size of the network. However the parameters which impacts our results the most is the composition of the access control policy.

Figure 2 also shows a comparison between the gain experimented in our simulations and the theoretical gain expressed in the previous section. This theoretical gain is represented by a dashed line. The comparison shows that simulation and theoretical results are very close.

When compared with the efficiency brought by the “topology optimization” (depicted with a gray line), our integrity scheme provides good results for big networks. As depicted in figure 3, the overhead becomes smaller as the size of the network increases (less than 3% with a 121 nodes network for both policies). However the overhead can be quite large for very small networks (around 18% with a 4 nodes network). Hopefully such a kind of networks is not very usual in real networks.

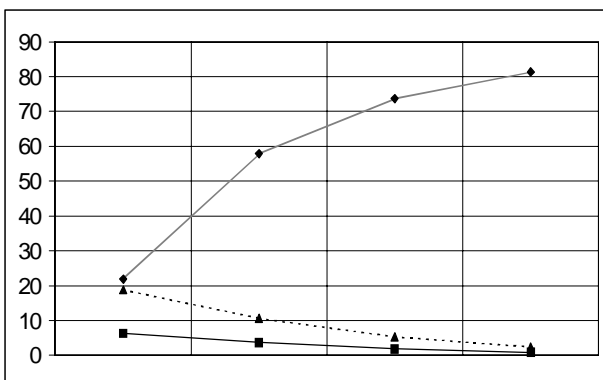


Figure 3. Criteria Global Efficiency (in %)

5. Conclusion

In this article, we introduce a scheme to improve the integrity of the access control process in a distributed access control environment. We show that this scheme can provide the security officer some insurance about the integrity of the access control process while limiting the overhead of such service. We show that our scheme is particularly interesting for big networks where the overhead can be as low as 2%. As a consequence, our criterion is especially interesting when used with a policy including a small ratio of deny rules in a large sized network.

This work could be usefully continued by integrating our scheme in a real access control management tool. This would allow us to test its efficiency in real world configurations.

Reference

1. Building Internet Firewalls, B. Chapman, E. Zwicky, O'Reilly & Associates, 1995.
2. Firewalls and internet security, repelling the wily hacker, B. Cheswick, S. Bellovin, Addison-Wesley publishing company, 1994.
3. Integrated Management of Network and Host Based Security Mechanisms, R. Falk, M. Trommer, 3rd ACISP'98 Conference, July 1998.
4. Firmato, A Novell Firewall Management Toolkit, Yair Bartal, Alain Mayer, Kobbi Nissim, Avishai Wool, IEEE Symposium on Security and Privacy, May 1999.
5. Filtering Postures: Local Enforcement for Global Policies, Joshua D. Guttman, IEEE Symposium on Security and Privacy, May 1997.
6. Policy-Based Management: Bridging the Gap, Susan Hinrichs, 15th ACSAC Conference, December 1999.
7. Management of Network Security Application, P. Hyland, R. Sandhu, 21st National Information Systems Security Conference, October 1998.
8. ns Notes and Documents, Kevin Fall, Kannan Varadhan, September 1999.
9. www.rennes.enst-bretagne.fr/~paul/acm.zip
10. An Asynchronous and Distributed Access Control Architecture for ATM networks, O. Paul, M. Laurent, S. Gombault, 15th ACSAC Conference, December 1999.
11. Distributed Firewalls, S. Bellovin, *login*., pp. 37-39, November 1999.
12. The Multilayer Firewall, D. Nessett and P. Humenn, NDSS'98 Conference, March 1998.