

# Specialized Solutions for Improvement of Firewall Performance and Conformity to Security Policy

V. Zaborovsky, A. Titov

Saint-Petersburg State Polytechnical University, Russia

**Abstract** - *Until recently the reasons for reduced efficiency and limited implementation of new security systems has been the insufficient performance of hardware that executes access control and the difficult analysis and configuration to conform with corporate security policy requirements. Without the use of specialized solutions that allow effective functioning of information security systems and their integration with other network applications, a well protected corporate network infrastructure is impossible. In this paper this thesis is considered from three perspectives: the choice of the distributed hardware platform to improve firewall performance; the description of security policy by means of an Organization Based Access Control mode, and automating the process of firewall rules formation based on high-level description of access policy requirements.*

**Keywords:** network access control, firewall, security policy, Organization Based Access Control.

## 1 Introduction

In a problem of increase of the information security efficiency it is possible to allocate for some levels: (1) hardware-network level; (2) function-system level. The choice of a hardware platform to design firewall appliances and proxy gateways concern to the first level. Description of security policy and access rules is the problems of the second level. Many researchers try to find perspective decisions which for the present have no a wide circulation in standard security appliances, but has a potential to improve protection feature of corporate network. At a level of the decisions concerning hardware platform, our accent is made on non-standard use of filtration technology when as a result of use of "stealth" mode the packet processing from nodes of a network are transferred to a level of communication channels. This approach allows us to use the high performance decisions that based on cascading of firewall CPU power without change of accepted routing policy and additional computing resources. In this case the increase in productivity is achieved also by dynamic character of filtering rules loading which are formed in reply to sequence of the asynchronous events connected with user's registration or unfriendly actions as DoS attacks. At a level of decisions of security system operation we use the approach which describes a security policy on the basis of division of the description of access rules and protection objects.

As a result the data from catalogues DNS (Domain Name System), Active Directory can be used for automation of process of the parametrical description of firewall rules. In such system dynamic character of access rules, allows to reduce the volume of protected parameters which in a wide class of situations reduces the general loading on firewall CPU. Therefore automatic formation of firewall rules based on use of role management mechanism provides: dynamism, simplicity and adequacy that are important for integration of information services used for the purposes of security management.

Taking into account all issues mentioned above we propose a specialized approach to address security appliances with basic functionality that can be automatically configured in accordance with a flexible access policy. The basic model has two definitive subsystems: (1) data processing environment, and (2) role-based user access policy to network information resources.

The first subsystem describes a network as a set of nodes with addressable interfaces employed for data exchange via a grid of links that forms the topology and the metric of network nearness. Within the standard network model all stages of information processing occur in network nodes, and lines are used only for data transmission. Thus, the basic functionality of network node is determined by two sequential stages of packet processing after their arrival from a communication line, namely, *store-and-forward*. The proposed model incorporates new functionality in the grid of communication lines, to add specific "stealth" nodes that logically belong to the lines of the grid because they do not use addressable network interfaces. This "stealth" node does not interfere with the existing routing policy and therefore does not change the network address space.

The second subsystem describes access policy and is used to configure "stealth" appliances in accordance with security and performance requirements. For this purpose a methodology is suggested that uses the security policy as the source for configuring the appliances. The methodology is based on Organization Based Access Control that makes the representation of access rules more similar to the security policy. The final firewall configuration is the result of the integration of network configuration and common access rules. The various benefits of the suggested methodology will also be considered.

## 2 Distributed Network Processor as High-Performance Firewall Platform

As the data transfer rate over communication lines increases and the protocol spectrum broadens, we are witnessing a growth in demands on the performance of the network processor (NP) employed in packet handling at network nodes. The architecture and specific features of operation of such processing engines have become the subject of a large number of studies [1-2]. Rather than drawing on a systematic analysis of the various specific requirements and design alternatives, most of these studies invoke the well-known results of the application of multi-processor architectures to increasing the speed of data flow processing. The solutions proposed to improve the functionality of the router now include firewalls, network address translators, means for implementing quality-of-service (QoS) guarantees to different packets flows and other mechanisms. Such implementations are based on several primary operations with packets: parse, search, resolve, and modify (Fig. 1). To implement all of these operations in real-time on a general purpose processor (GPP) often becomes infeasible due to performance requirements. This issue motivates solutions where the packet-processing functionality of the NP is implemented in specific pooled and pipeline hardware. Such a decision restricts flexibility. *The complex nature of packets operations favor software based implementations on a GPP.* To address these conflicting issues and organize the stages in packet processing, a relatively new *store-process-and-forward* scenario is proposed.

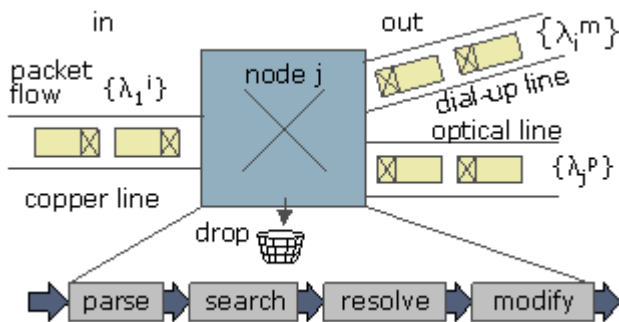


Fig. 1. Basic packet processing scenario

In general, all solutions may be separated in two classes. Grouped in the first class are the solutions aimed at boosting the router's performance. The main parameters governing such operations are the packet destination addresses, and, therefore, the solutions chosen are directed at accelerating the data search in the router lookup tables. The second class of solutions involve the implementation of procedures without routing decisions: packet classification, data processing, providing the required QoS, bandwidth allocation, and so on. In principle, this separation of the handling processes permits one to break up the network appliance performance between several types of distributed nodes that belong to one virtual transport connection. So, if a packet operation among such

nodes occurs without the use of a routing decision, they can be functionally assigned to communication lines. This approach modifies the basic network scenario from store-and-forward to a process-store-and-forward concept. This concept offers a solution to providing the necessary flexibility by keeping the basic routing operation and adding new functionality without changing network topology and computation power.

In practice, this concept actually uses the OSI model to provide several control layers. At each layer data structures are controlled by specific rules. The corresponding control rules can be broken down into the following stages: (1) collection of data to be transmitted through the network; (2) configuring a structure to quantitatively determine the volume of the data to be transmitted; (3) attaching to the data a special header specifying the set of parameters to be used in handling the packet in network nodes; (4) formation of a frame meeting the requirements of the communication line hardware; and (5) frame transmission over the communication line connecting two network nodes. The above stages are prone to various malicious actions which are capable of interfering with the standard procedure of packet transmission or of substituting packets on the way from their generation. A protective mechanism can be implemented by several means, which can be divided into methods of packet filtration, and cryptographic processing. The first group of methods protects the network application by means of special firewall rules [1]. In common configurations the firewall does not become an end point of packet transmission and has to be installed in the network segments crossed by packet flows. Routing policy does not have to change to accommodate the firewall, but if network metrics do not change due to the filtering interfaces that have no physical and network address, the situation can be changed. The efficiency of address-less firewall processing is fully determined by the algorithmic features of used filtering engine and available processing power. In the case of firewall systems the factors of particular significance are: the number of simultaneously processed virtual connections, QoS requirements, and the acceptable level of packets delays. Because the transmission of packets is executed in an asynchronous mode, i.e., it is initiated independently by each node, the number of logical connections passing through the node is a random quantity obeying a fractal distribution function [1]. In this case standard firewall architecture does not fully determine its performance. The optimum number of packet processing stages depends on the actual characteristics of the security requirements. A new type of distributed firewall appliance can be based on the separation of packet processing into base and additional operations. Among the base operations is packet routing. Other operations which do not demand routing of packages define the set of additional processing options. Such options can be executed in nodes, whose network interfaces have no IP and MAC addresses. Application of this approach to firewall design implies that one part of packet processes is carried out in a parallel mode at network node and other part in sequential pipeline mode in

a “stealth” NP node adjoining the addressable network node (Fig. 2).

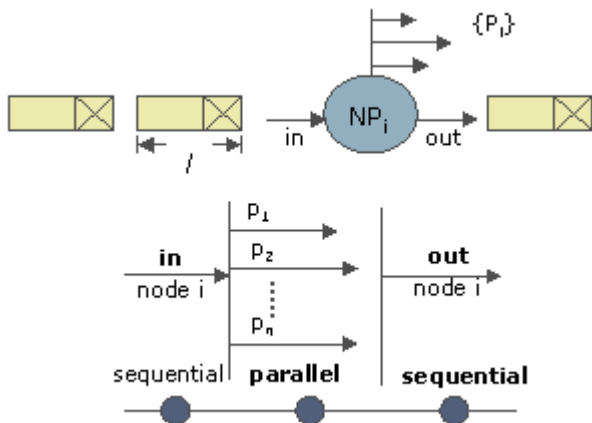


Fig. 2. Firewall as a network processor (NP)

The development of a stealth mode for a distributed firewall appliance becomes possible because such devices do not act as sources or destinations of network packets in most of their operational regimes. Therefore, the network interfaces of these devices may have no physical or logical addresses altogether and, hence, transmission of IP packets or MAC frames through them becomes similar in character to their passing through a hub or cable line segments used in packet exchange. To operate successfully, a firewall should work like a “sophisticated” parallel bundle of network cables or a transparent but secure logical channel between the network nodes. The next step of firewall decomposition is based on sequential-parallel-sequential stages in packet-handling processes which offer a possibility to cut packet delays in the packet reception and processing mode. Operational sequencing in the second mode can be integrated into a specialized virtual pipeline cluster and spread out between “stealth” nodes (Fig. 3). In this scheme, firewalls can either “bridge” or “route” traffic. In the first case, the firewall functions as a layer-2 network bridge with IP transparent or “stealth” interfaces. This means that each interface has a MAC address but the network or IP address space is the same on both sides of the firewall.

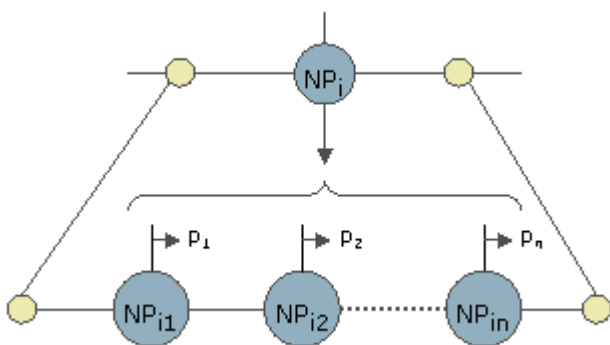


Fig. 3. Virtual pipeline cluster

This method of concealing the network address of information protection devices, on one hand, provides the conditions necessary for execution of the protection functions, while on the other, because of the packet processing device network interfaces having no addresses, no changes in the network connection topology and in the already accepted packet routing policy are required. Security devices based on “stealth” address technologies have a number of advantages not only due to their concealed functioning but also from the standpoint of the scalability of performance and enhanced reliability of operation. The improved performance originates from the use of sequential/parallel character of the network traffic employed, where independent logical connections form through pipeline transmission of packets with definite addresses of message sources and receivers. Operation with network devices based on IEEE 802.3 Ethernet technologies in the stealth mode permits packet processing in the kernel of the built-in operating system without using the TCP/IP protocol stack. This method of processing reduces the packet buffering delay fluctuation level, which likewise improves concealment of the location of protection devices.

### 3 Configuration of Firewall

A firewall is a network processor which executes access control in networks. It’s among a large number of various devices that are employed for security policy support in organization. In modern networks a security officer must configure all of these devices in compliance with current security policy. Whereas he often doesn’t have enough knowledge about the specificity of an information security (IS) component (in particular today’s complex firewalls) or he doesn’t have enough time to explore and implement every functional detail. As a result these details become unnecessary and even burdensome. *To make the work of the security officer more effective it’s necessary to have a model for each device that would represent a part of its functionality. This part must be complete enough to work with the device as an IS-component while all the unnecessary details must be hidden.* The building of such a model for firewalls is considered further.

There is a feature that is common for all firewalls: they execute an access policy. Access policy defines who (user) is allowed to access to a resource (network services). In general, an access policy is executed by various access control devices (ACD). In common representation the main function of ACD is to decide whether a *subject* should be permitted to perform an *action* with an *object*. A common access rule:

**Mr. Black is prohibited to read www.youtube.com.**

In the above example Mr. Black is a subject; HTTP service on www.youtube.com is an object. Reading is an action. So the configuration of ACD consists of common access rules which reference to subjects, actions and objects.

*Although a firewall is an ACD and must be configured with common access rules as source, each one uses its own specific configuration language. The language often reflects*

the features of firewall's internal architecture and traditionally is represented by a set of firewall rules. Each rule has references to host addresses and other network configuration parameters. The verbal description of a firewall rule:

**Host with IP address 10.0.0.10 is prohibited to establish TCP connections on HTTP port of host with IP address 208.65.153.238.**

*It's obvious that just ACD may be a common representation of firewall as an IS-component.* The main complexity of this approach is to know how firewall rules could be obtained from common access rules.

Each firewall vendor reasonably aims at increasing its sales appeal while suggesting various tools for convenient editing of firewall rules. However, so far the problem of obtaining firewall rules from common access rules is not resolved. Moreover, this problem has not been paid much attention.

*The most obvious issue concerning the problem is that additional information is necessary beyond access rules in order to obtain firewall rules.* This information is about the configuration of network services and the parameters of network protocols that are used for data exchange - *network configuration*. It can be mainly stored among the descriptions of subjects, actions and objects. An example:

**Mr.Black: host with IP-address = 10.0.0.10;  
www.youtube.com: HTTP service on host with  
IP-address = 208.65.153.238.**

Thus, final firewall rules can be obtained via the addition of object descriptions to access rules. It should be noted that even for small and especially for medium and large enterprises it is necessary for the network configuration to be stored and managed separately from the security policy. The suggested approach allows us to achieve this goal: the security officer can edit access rules with references to real objects while the network administrator can edit the parameters of network objects.

It should also be noted that there is no need now to specify any fixed rules regarding how network parameters could be associated with objects. For instance, HTTP port may be a parameter of object or it can be a parameter of action. A criterion is that the most natural representation of access policy must be achieved. A concrete, probably subjective decision may vary from one system to another [2].

Parameters of network objects can be automatically retrieved from various data catalogs. DNS is the best example of a world-wide catalog which stores network addresses. Microsoft suggests network administrators the powerful means, Active Directory [3], to store centrally information about network objects. Integration with the above technologies relieves the security officer from much work as he must only specify the correct name of an object while forming firewall rules.

### 3.1 Role Based Access Control and Or-BAC Model

*The access rules which use the notions of subject, action and object are the basis of access control policy but they are not sufficient alone to implement complex real-world policies.* As a result, new approaches have been developed. One of them, Role Based Access Control (RBAC), uses the notion of role - an abstract notion which is related to access control policy only. A role doesn't have fixed associations with the real objects of information system. A role can replace subject in access rules. As an example, remember the roles of system administrator and unprivileged user that are commonly used while configuring various systems. It's obvious that a role must be associated with some subjects as only rules with subjects can be executed. Thus, during the realization of security policy the access rules for all the subjects must be specified. It must be done in two steps. First, roles must be created and access rules must be specified with reference to these roles. Then the roles must be associated with subjects. So, a role can be used in order to group subjects and for more clear access control policy representation.

The Or-BAC model [4] expands the traditional model of Role Based Access Control. It brings in the new notions of *activity* and *view*. The notion of activity corresponds to action and its meaning is analogous to the meaning of role for subject. The notion of view corresponds to object. "Entertainment resources" can be an example of view and "read" or "write" can be an example of activity. Thus, the notions of role, activity and view form a new abstract level of access policy. Or-BAC allows the specification of access rules on an abstract level using roles, activities and views. Therefore they are named abstract rules.

Rules with subjects, actions and objects are called concrete rules.

Consider an example of abstract rule:

**User is prohibited to read entertainment resources**

"User" - role, "read" - activity, "entertainment resources" - view.

### 3.2 Configuration Methodology

The use of Or-BAC model for configuring a firewall makes it possible to divide the configuration process into several stages (Fig. 4):

*Stage 1.* The security officer must analyze all the requirements of security policy and write down all the abstract access rules which follow from the security policy. This stage is not so hard to implement as formulations of abstract rules are similar to sentences being used in official documents.

*Stage 2.* The only abstract rules are not sufficient. The model must be completed with real objects. These objects must be associated with roles, views and activities. The security officer must classify and describe all the resources and users in own organization.

*Stage 3.* Information about the network configuration must be available in order to get the final firewall rules. The network configuration may be specified manually by some expert (such as the network administrator, not necessarily the security officer) as a part of objects descriptions.

*Stage 4.* The process of obtaining firewall rules for a concrete firewall may be implemented automatically (the module called Or-BAC network adapter). There must be the abstract rules, the objects associations and the network configuration for input into the algorithm. The firewall rules are the output of algorithm. Some additions to the main algorithm may be required while working with different firewall models but if the models are similar then these additions will not be significant.

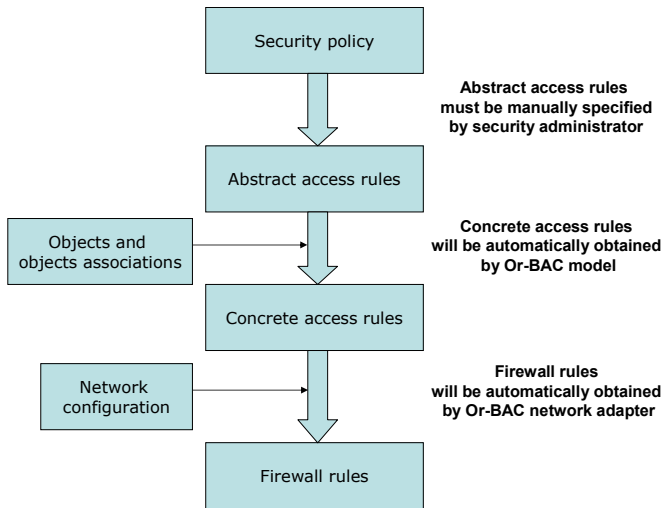


Fig. 4. Obtaining firewall rules from security policy

### 3.3 The Benefits of the Suggested Methodology

#### The Decrease of Probability of Human Error

A feature of the suggested methodology is that the process of firewall configuration is divided into several stages. The most informal first stage is made quite simple for the security officer due to the suggestion of instinctively understandable notions to specify abstract access rules. Thus, the abstract rules can be easily specified, analyzed, edited and compared with an informally specified security policy. This will reduce the probability that a firewall will not work in compliance with the security policy because of a security officer error.

#### Ready Access Policy

Most abstract rules can be initially stored in prepared templates. For example, an abstract rule that a user can't read entertainment resources is quite common. It will be certainly employed by most of security officers. So if they have a prepared template with such common rules, then they must only describe the real objects and the network configuration of their own organization in order to obtain firewall rules.

### Dynamic Firewall Configuration

The network parameters of objects may change; for instance, a user may log on any computer so his IP address may change. However, the access rules for this user must be consistent. Thus, the configuration of firewall must be changed dynamically. The process of such a dynamic configuration requires the consideration of common access rules which are the source of firewall rules. The suggested methodology initially considers common access rules so the evolution from static firewall configuration to dynamic one can be more easily developed.

### Reliable Configuration of System of ACDs

The suggested methodology uses the notion of an abstract ACD which corresponds to a firewall or another real ACD (for instance, a file access controller of operating system). The abstract ACD represents the functionality of the real ACD that is only related to information security. Thus, very different real ACDs can be represented by similar abstract ACDs. This enables easier consideration of an information system that employs many different ACDs. In such a system IS-components may interact with each other while pursuing the common goal: the execution of a complex security policy. As each ACD has its own configuration then it's possible that configurations of different ACDs will not correspond to each other, of course this would create a conflict. If such conflicts emerge and remain unnoticed then the behavior of the system may be unpredictable and may not appropriately enforce the security policy. Unnoticed conflicts can be avoided if the configurations of all the ACDs have a single source. A single distributed ACD must be considered. It may be a difficult task since the ACDs are very different. The use of similar abstract ACDs can provide a better approach to this problem.

## 4 Conclusion

1. Perspective ways to increase the security of computer networks require the development of high-performance platforms for firewalls and the deployment of access policy description models that allow of automation of firewall rules adjustment.
2. The high-performance security appliances can be developed on the basis of distributed computing platforms which are implemented as a set of "stealth mode" network processors.
3. The "stealth" operation feature improves firewall scalability and offers the possibility of integration of additional processing procedures into the standard network environment that not changes the current routing policy.
4. Application of the "stealth" technology cuts the costs of network upgrading and permits of redistribution of the required CPU power among various network devices. Proposed approach provides the possibility to increase the performance of network nodes and overall system reliability.
5. Each firewall is required to work in compliance with a corporate security policy. This policy cannot be considered separately from security appliances, methodology of proper firewall configuration and an informally specified security

requirements. The methodology also employs Or-BAC model which helps to translate high-level abstract access rules to low-level firewall features in order to obtain the final access configuration.

6. Firewall configuration can be largely automated based on specifying high-level access rules and parameters of corporate DNS, AD services. Suggested methodology can be easily implemented due to consideration of role-based information access models and characteristics of network appliances.

## 5 References

[1] V.S. Zaborovsky, Y.A. Shemanin, Jim A. McCombs, A. Sigalov «Firewall Network Processors: Concept, Model and Platform», Proceedings of International Conference on Networking (ICN'04), Guadeloupe, 2004.

[2] F. Cuppens, N. Cuppens-Boulahia, T. Sans and A. Miège, A formal approach to specify and deploy a network security policy, Second Workshop on Formal Aspects in Security and Trust (FAST). Toulouse, France. August, 2004.

[3] Windows Server 2008. Active Directory.  
<http://www.microsoft.com/windowsserver2008/en/us/active-directory.aspx>.

[4] A. Abou El Kalam, R. El Baida, P. Balbiani, S. Benferhat, F. Cuppens, Y. Deswarte, A. Miège, C. Saurel et G. Trouessin, Organization Based Access Control, IEEE 4th International Workshop on Policies for Distributed Systems and Networks (Policy 2003), Lake Como, Italy, June 4-6, 2003.