Visual Firewall Rule Builder

Eugen PETAC, Ph.D.*, Tudor UDRESCU**

*"Ovidius" University, Constantza – Romania

Abstract:

The paper that is being submitted deals primarily with the implementation of firewall technology on Linux based systems. It focuses on the features of the visual medium for the creation and management of firewall rules, ***V**isual **F**irewall **R**ule **B**uilder – **VFRB***. **VFRB** has been written in Java and has a modular construction. The program consists of an object-oriented graphical user interface and a policy compiler for *iptables*. In **VFRB**, a firewall security policy is a viewed as a set of rules, where each rule is made up of abstract objects that represent real network objects and services (hosts, networks, protocols and so on). **VFRB** helps maintain a database of objects and enables the editing of firewall policy using simple drag and drop operations.

1 Firewall Survey

Network security has become an important issue for companies and private users [1],[3]. Although the Internet has evolved into an extremely powerful tool for the distribution and gathering of information, at the same time, its use is not risk-free. Information theft, computer related criminal activities, data loss, are all potential dangers, that no organization or individual can ignore. In such a dangerous environment as the Internet, a firewall is the best method to safeguard a network. Although a satisfactory level of security can only be achieved when a firewall is used in conjunction with other types of protection, this technology should be at the very center of the security plan for any system directly connected to the Internet [9].

A firewall is a structure intended to keep a fire from spreading [2], [5]. Internet firewalls have the same purpose, as they are designed to protect a LAN from unauthorized access from outside. As a matter of course, the term *firewall* is used to identify a secure and trusted machine that sits between a private network and a public network. This machine is directly connected to the Internet and is configured to implement a security policy [8]. The security policy consists of a set of rules that determine which network traffic will be allowed to pass and which will be blocked or refused.

One of the most common types of firewall is a filtering firewall. A packet filtering firewall works at the network level of the TCP/IP protocol stack. Data is only allowed to leave the system or the private network if the firewall rules allow it. As packets arrive from the public network they are filtered by their type, source address, destination address, and port information contained in each packet.

2 Configuring a firewall in Linux [5]

There is no need for special software in order to create a filtering firewall in Linux. The only requirement is the presence in the system of a kernel that supports packet-filtering technology. All Linux distributions based on the 2.4 series kernels have support for the latest generation of filtering firewall, called *netfilter*.

2.1 Iptables

The *iptables* utility is used to configure the rule sets of a *netfilter* based firewall. The most significant aspect of iptables is that it is extensible. This means that the functionality of the software can be extended without recompiling it. The secret lies in the use of shared libraries. The utility comes with some standard extensions that provide additional features for *iptables*. More extensions can be added in the future , which makes *iptables* a very versatile and flexible program.

The general syntax of the iptables command is:
iptables command rule-specification extensions

For more information about *iptables* commands and parameters please see http://www.netfilter.org/ and http://www.iptables.org/.

3 Visual Firewall Rule Builder based on *iptables*

The *iptables* firewall administration program included in the Linux distributions lacks a graphical interface, therefore the process of generating or even modifying a firewall security policy is lengthy and difficult, involving the manual editing of shell scripts. It is obvious that building a firewall in this manner is a time consuming process and places on the shoulders of the system administrator the additional burden of checking the code's correctness. The development of a good security policy for a firewall is a fine art, and the whole process is a challenge to even the most capable security specialist. There is no need to complicate things even further by forcing the user to deal with the rather unpleasant task of manually editing the scripts that implement the security policy. Unfortunately, the *iptables* program does not offer any solution to this problem, facing the classic problem of all Linux applications: the absence of a graphical interface to facilitate the user's interaction with the software. Because of this, many inexperienced users, who wish to protect themselves against the dangerous aspects of Internet use, are hesitant about implementing a Linux based firewall or, if not, they create a firewall which does not offer an efficient protection. This happens because the „classic" construction and maintenance method is too complicated for a user that just migrated towards Linux from a Windows environment.

Using the Java language, the VFRB (Visual Firewall Rule Builder) application has been created to solve this problem and to offer a visual environment for the building and management a firewall's security policy.

3.1 The Building of the application

Visual Firewall Rule Builder (**VFRB**) is composed by an object-oriented user interface (GUI), a rule compiler and a manager for installing the rules of the firewall. In **VFRB** a firewall policy is seen as a set of rules, each rule being formed from a series of abstract objects that represent real network objects (hosts, networks, interfaces, address ranges), and services that use one of the following protocols (IP, ICMP, TCP or UDP). The application communicates with a database that contains all the objects involved in the construction of a filtering rule and allows the editing of the security policy through drag-and-drop type operations.

The structure of the application is modular, the user interface, the rule compiler and the module for installing the firewall can all be modified independently. Consequently, the application can

easily be upgraded to incorporate any changes that may occur in the Linux firewall. It also is possible to add more compilers to deal with other firewall administration commands, like the older *ipchains* or *ipfwadm*.

The core of the application is the Network Objects Database, which is a centralized storage of information about the network objects involved in the building of firewall rules. For example a host object would contain information about the name, number of interfaces, their logical and physical addresses, and other important information. For a network the essential information are the network's address and the subnet mask. Currently the Network Objects Database is stored in plain text format in independent files for each category of objects.

The user interface of **VFRB** was developed using the Java language. All graphical components, that form this interface, can be found in the *Swing* (JFC) package. This technology offers several advantages over the older AWT (Abstract Window Toolkit) and has become the de facto standard in developing user interfaces with Java. Besides the pleasant appearance of the GUIs made using *Swing*, this technology based on *Java Beans* provides a comprehensive library that contains all the necessary graphical components for the development of a modern and functional interface.

VFRB operates in two modes. The browser mode in which the user can navigate through the database and modify its contents, and the rule edit mode. In the rule edit mode, the user can visually create rules using the network objects as building blocks. The GUI allows viewing, creating and editing the network objects. The techniques of drag and drop are used widely to construct firewall rules. This allows for speedy creation of a firewall, and for easy management of firewall rules. The navigation is done by moving the mouse or by using keyboard shortcuts. The objects are organized on screen in a tree-like hierarchy in the left pane of the main window while object properties and rules appear in the right side of the window depending on the current mode of the application. The application represents a firewall security policy in the form of a list of filtering rules. Each rule has standard fields, or rule elements such as Source, Destination, Service, Action, Options, Direction and Comments. The Source and Destination hold references to network objects such as hosts, networks, address ranges. The Service holds reference to a service (IP, ICMP, UDP or TCP). The references are created using the drag and drop technique. The user selects a network object and places it in the appropriate field.

The user can manage the rules, change their position inside the policy, delete selected rules or insert rules at specified locations inside the rule set. The configuration of a firewall can also be saved for later use. Also, previously saved configuration can easily be loaded from a storage medium. A standard Copy/Paste mechanism for the objects that compose a rule is also supported.

The rule management part of the application is one of its strong points, as the position of the rules inside the security policy is vital to the success of a firewall. Filtering firewalls operate on the principle of "the first rule that matches wins", therefore rules should be ordered from the general to the specific. Due to this reason the program treats the policy as a chained list where the filtering rules are the elements. With VFRB, the administrator can quickly change the parameters of a firewall configuration. Since firewall creation and management is based on trial and error and on quick responses to emerging dangers, in this respect, VFRB can prove a powerful tool for securing a network. Besides the graphical interface, **VFRB** also includes a rule compiler that generates a standard shell script based on the rules defined by the user in the graphical interface. The syntax used is that of *iptables*. The program also contains a mechanism for installing the newly created security policy as a background operation.

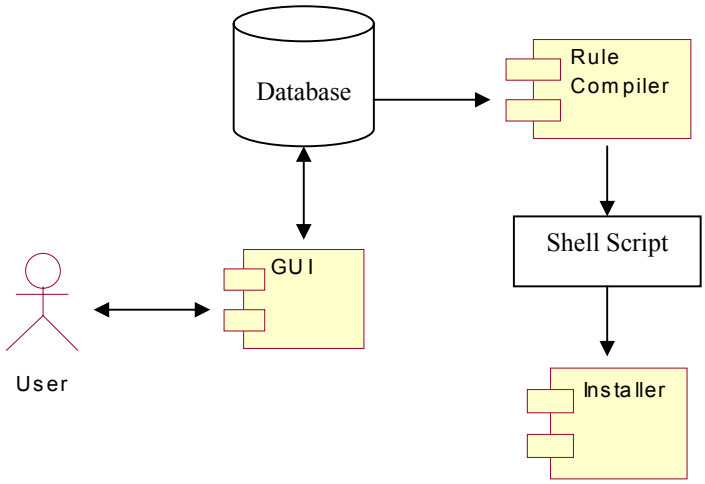

Figure 3 - Architecture of the VFRB

4 Developing a security policy

In order to begin the development of a security policy, the user must first define the objects that model the network that he wants to protect. To speed up the process, the program has a database that already contains most of the services or network objects that can be encountered. Each service or network type object can be modified and also new objects can be added to the database. After defining the objects we can proceed to the next step, which is the creation of the policy by constructing the rules.

The process starts by adding a rule at the beginning of the policy; it continues by inserting more rules and populating them with objects. The building of a rule is done through the simple drag-and-drop operation. An object is selected from the tree that displays the content of the database from the left panel of the application and is placed in the appropriate box in the right panel. To do these operations, the application must be in the rule-editing mode. VFRB checks the type of object and allows only the creation of valid references. For example it is not possible to place a service type object in a Destination box or a network type object in a Service box. When a reference between an element of a rule and an object of the database has been created, the properties of the object can be modified independently of the properties of the object from the database.

5 Conclusions

This paper is a presentation of the features of a visual medium for the creation and management of firewall rules – VFRB. **VFRB** is composed of a user interface, a rule compiler, which generates a executable script based on the rules defined in the interface, and a manager for installing the security policy.

The application distinguishes itself especially through flexibility and the ease in installation and use. Most similar programs are either too complex for the needs of a common user, or require the purchasing of a license.

References

[1]. BRENT D. CHAPMAN, ELIZABETH D. ZWICKY : Building Internet Firewalls, O'Reilly 1999
[2]. STEVE FRAMPTON : Linux Administration Made Easy , O'Reilly&Associates 1999
[3]. MARCUS GONCALVES : Firewalls Complete, The McGraw-Hill 1997
[4]. TREVOR KAY : Linux+ Certification Bible, Hungry Minds Inc., 2001 pp.462-480
[5]. OLAF KIRCH, TERRY DAWSON : Linux Network Administrator's Guide, O'Reilly 2nd Edition 2000
[6]. VADIM KURLAND : Firewall Builder Tutorial, www.fwbuilder.org
[7]. SANDRA A. MOORE, TAMMY FOX : Red Hat Linux 8.0: The Official Red Hat Linux, Reference Guide, Red Hat, Inc. 2002
[8]. E. PETAC, D. PETAC : Technical Principles of Information Security, Matrix Rom, Bucuresti, Romania, 1998.
[9]. ROBERT L. ZIEGLER : Linux Firewalls (2nd Edition) by Robert Ziegler ,Teora 2001