International Journal of Engineering & Technology, 7 (4.44) (2018) 237-241

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Use of Virtualization Technology for Linux Firewall Implementation in Teaching Computer Network

Yuri Ariyanto[1], Budi Harijanto[2], Yan Watequlis S.[3], Awan Setiawan[4], Erfan Rohadi[5]

[1,2,3,5]Information Technology Department, State Polytechnic of Malang, Indonesia
[4]Electrical Engineering Department, State Polytechnic of Malang, Indonesia
**Corresponding author E-mail: yuri@polinema.ac.id*

Abstract

This paper describes the use of virtualization technology in the linux network teaching process in linux firewall implementation. The problem that is often faced by teacher and student is the limitation of computer devices in implementing network topology design. The use of virtualization technology can be used as a device to create a virtual network laboratory, which is an implementation of real laboratories and network simulation software that is useful to explain the concept of computer network management. With this in mind, it can provide an overview of project characteristics, especially the implementation of linux networks in the area of virtualization technology. Linux network virtualization implementation to help the teaching process used netkit. Netkit can be used as a linux network simulation, where each virtual machine host has a linux operating system, and can implement a computer network based on the design of a real network topology. The method used to implement the Linux firewall by creating a virtual laboratory, where virtual laboratories are built based on the real network topology design. Testing is done on a virtual laboratory by implementing a network topology design and running based on a test scenario.

.

Keywords: *Virtualization Technology ,Netkit, Linux, Firewall, Design Network*

1. Introduction

The development of computer virtualization technology at this time, experienced a good development as a computer network learning media. With the emergence of network emulators, it can help teachers and students understand the learning provided. In computer network learning, teachers use network simulation for network topology implementation [1]. Network emulators are used by administrators, researchers, teachers for implementing real network designs to network emulators, making it easier to test scenarios that have been created [1]. In network learning, network emulators can be used as test environments in a virtual laboratory. To make it easier for students to understand network material, the teacher uses virtualization technology teaching media, to help students in lab work [2].
In this paper explains how to implement a real network topology design in a virtual laboratory, to help teach iptables firewall on linux. Virtual laboratory testing is done by running a network scenario that has been created, to find out if the network scenario is running.

2. Supporting Theory

2.1. Virtual Machine Concept

Virtual machines are built using virtual machine monitor technology (VMM). VMM hardware is a complete computer system which consists of software and hardware [3]. The use of virtualization technology is very popular when hardware is very expensive.

2.2. Network Emulator

To facilitate the implementation of network topology in the laboratory, network emulators are used for implementation. With network emulators it will facilitate understanding in network topology implementations, because network emulators have already installed network operating systems, which can implementation network topology [1].

2.3. Netkit

Using Netkit virtualization technology, you can experiment with real network topology, because network hardware is available. Implementation of virtualization using netkits will not interfere with the running network system. This technology is very helpful for researchers and educators in conducting experiments before being implemented on real networks, before being widely implemented in the network topology that has been created [4-8].

Copyright © 2016 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

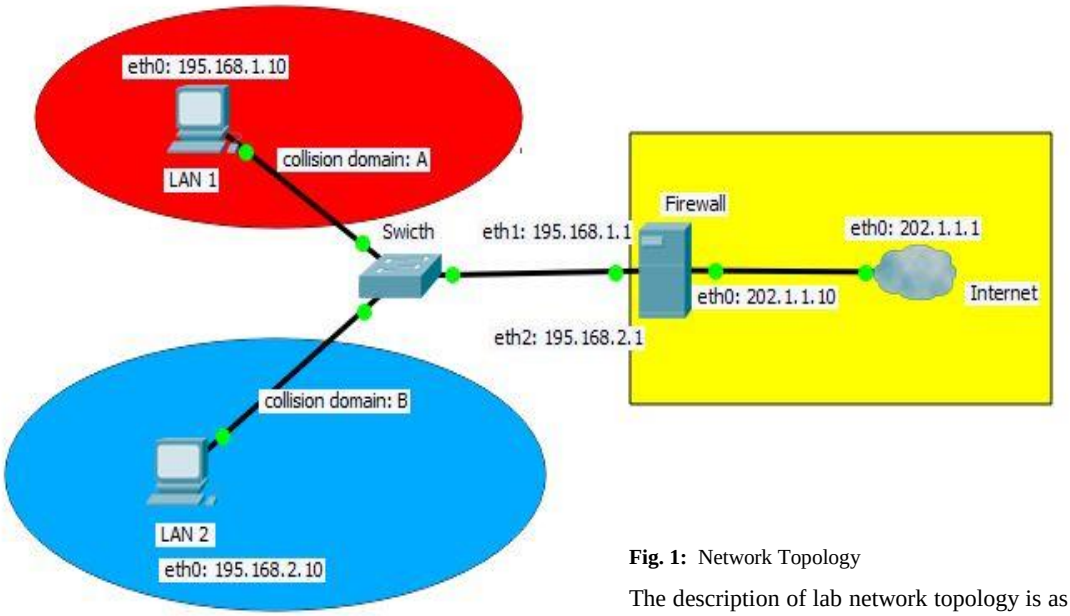

Fig. 1: Network Topology

The description of lab network topology is as follows. A complete description of the network topology is shown in table 1

Table 1: Design Network Topology

No	Computer Symbol	Network Card	Host Address
1	LAN 1	Interface eth0	195.168.1.10/24
2	LAN 2	Interface eth0	195.168.2.10/24
3	Firewall	Interface eth0	202.1.1.10/24
	Firewall	Interface eth1	195.168.1.1/24
	Firewall	Interface eth2	195.168.2.1/24
4	Internet	Interface eth0	202.1.1.1/24

3.3. Analysis System Requirements

System requirements analysis is used to explain the needs of software and computer hardware used in the implementation of netkit virtualization technology. Analysis of system requirements are shown in table 2.

Table 2: Analysis System Requirements

No	Name	Specification
1	OS Linux	Debian 7
2	Processor	Intel i7
3	RAM	4 GB
4	Netkit	Netkit-2.8, Netkit-filesystemi386 and Netkit-kernel-i386

4. Virtual Laboratory Scenario

At this stage, the lab will be described. The virtual lab iptbales firewall is built with virtual machines using netkits. Virtual machine lab netkit built using a real network topology design shown in Fig. 1. Scenario testing, used virtual lab netkit machines in its implementation, explanation of the implementation as follows

4.1. Create and Configure file lab.conf

The lab.conf file is created to create a network scenario, adjusted to the real network design scenario that has been created. The lab.conf file contains the computer needs and specifications according to the network topology created. The lab.conf configuration file is shown in Fig. 2.

2.4. Linux Iptables

In the Linux distribution, the iptables firewall is used. Iptables is a firewall software that is included with most Linux distributions by default [9-10].
The iptables architecture processes packet processing rules into tables based on the input, ouput and forward rule functions, the rule functions in each have a chain to filter access to and out of the network.
The rules in iptables consist of matches (used to determine which packages to apply to rules) and targets (which determines what to do with a suitable package). The iptables firewall application on the Linux operating system runs on the network layer in the OSI Layer model [9,11].

3. Methodology and System Design

3.1. Methodology

This research, using research methods in the following stages (1) needs analysis is by collecting various information related to computer network learning media. (2) network topology design based on needs analysis (3) create test scenarios on network topology (4) evaluation of trial results based on testing scenarios.

3.2. Design Network Topology

Virtual laboratory experiments implementing ip-tables firewall is depicted on the network topolo-gy implemented on Netkit, as shown in Fig. 1.

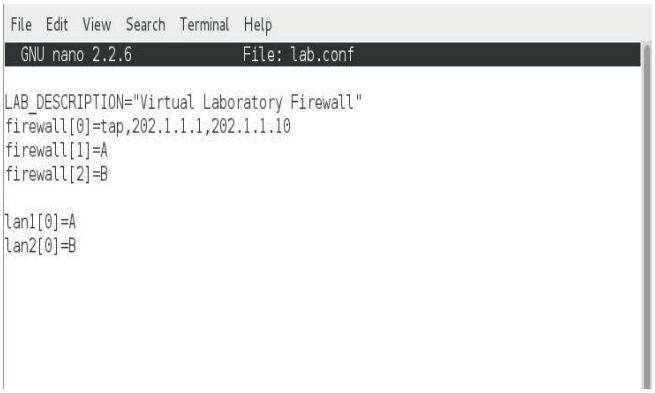

Fig.2: File lab.conf configuration

Configure in the lab.conf file based on a real network topology design. The lab.conf file is used for virtual lab implementation. Where in the lab.conf all network devices will be connected to the connection between the NICs of each computer using virtual domain collosion. The domain collosion is a symbol that connects virtually the NIC to the Netkit emulator. Virtual laboratory implementation. The firewall consists of virtual machines as firewall computers, LAN1 and LAN2.

4.2. Create and Configure Virtual Machine (VM) in lab.conf

At this stage the researcher will configure the .startup file in Netkit. This file is needed for configuration and services that will be run when the virtual machine is run on a virtual lab.

- Configure VM Firewall

Create and configure a firewall.startup file show in Fig. 3.

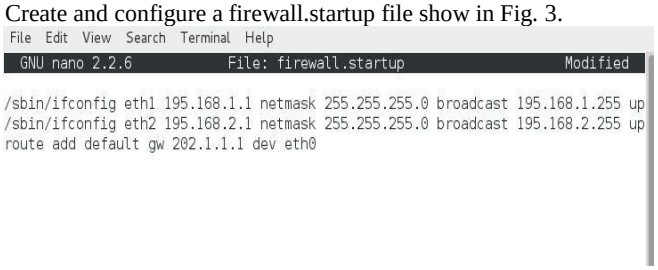

Fig.3: firewall.startup

In Fig. 3, we describe the configuration of ip address eth1 195.168.1.1/24, eth2 195.168.2.1/24 and ip gateway 202.1.1.1 for firewall computer.

- Configure VM lan1

Create and configure a lan1.startup file show in Fig. 4.

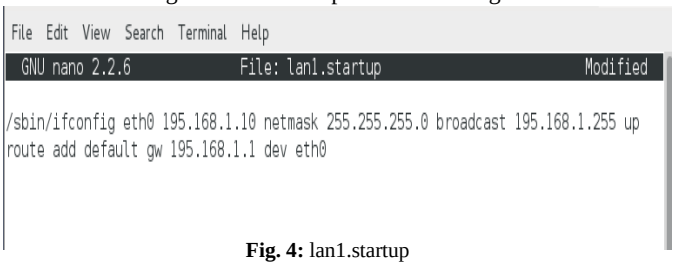

Fig. 4: lan1.startup

In Fig. 4, we describe the configuration of ip address eth0 195.168.1.10/24 and ip gateway 195.168.1.1 for lan1 computer.

- Configure VM lan2

Create and configure a lan2.startup file show in Fig. 5.

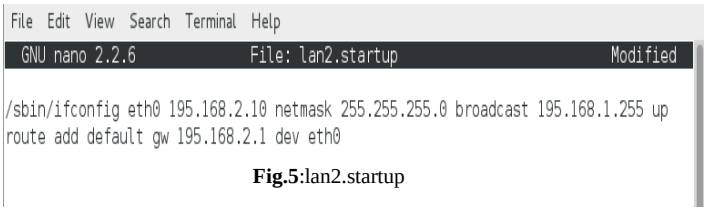

Fig.5:lan2.startup

In Fig.5, we describe the configuration of ip address eth0 195.168.2.10/24 and ip gateway 195.168.2.1 for lan2 computer.

4.3. Virtual Lab. Implementation

Implementation of the network topology design in Fig.1. at netkit emulator shown in Fig. 6.

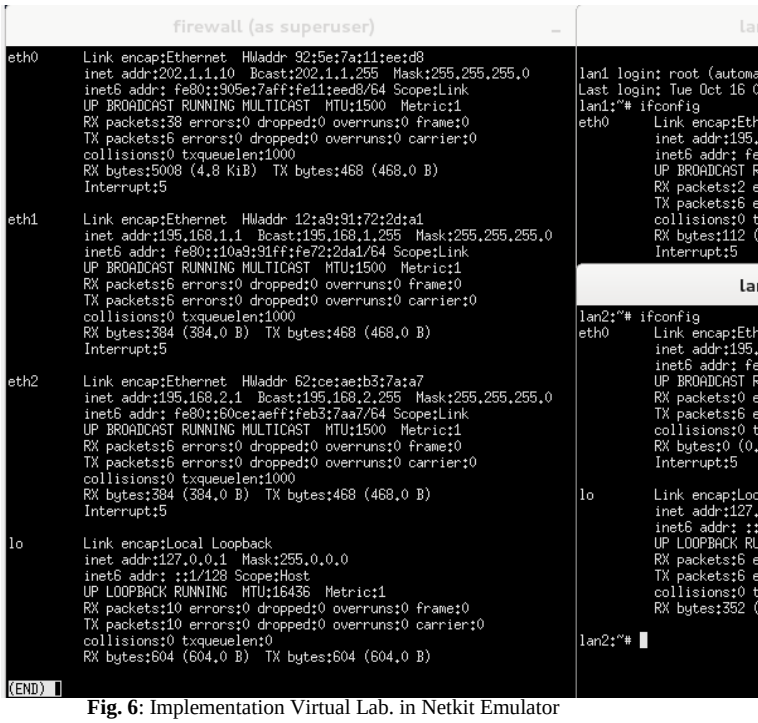

Fig. 6: Implementation Virtual Lab. in Netkit Emulator

In Fig. 6, The results of the virtual laboratory implementation are shown based on the real design of the network topology, the design is tested.

5. Testing and Analysis Virtual Lab.

5.1. Automation of Firewall Iptables Rules

At this stage the firewall is made to iptables VM firewall, by creating a firewall.sh file in # /etc/ init.d /firewall.sh.With the Linux command #pico /etc/init.d/firewall.sh

Then the firewall iptables rule is created in the firewall.sh file, shown in Fig. 7.

Fig.7: Scenario Rule Iptables Firewall

In Fig.7. The iptables firewall rule is created as follows:

- Block icmp protocol VM lan1 to VM firewall
- Accept icmp protocol VM lan2 to VM firewall
- Accept and open port remote ssh VM lan2 to VM firewall.

For the firewall.sh file, to be able to be used with full access it needs to be given an additional configuration #chmod 777 firewall.sh. The result of the firewall.sh service call with the command # /etc/init.d /firewall.sh start, is shown in Fig. 8.

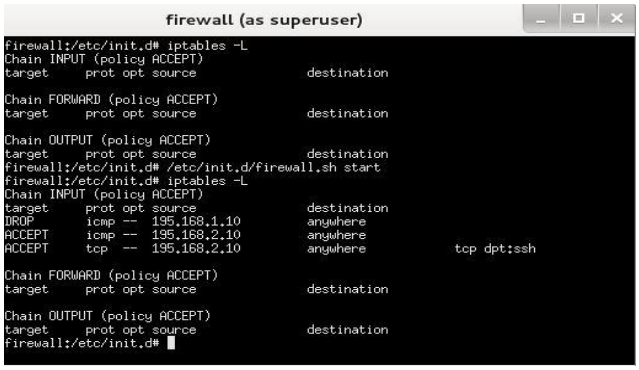

Fig.8: Running Service firewall.sh

In Fig. 8 it is shown, the easy writing iptables firewall rules, because writing is do automatically by activating the firewall.sh service. With this service, users do not write one by one firewall rules in the linux command line.

5.2. Firewall Scenario Testing in Virtual Lab.

In Fig. 9, the icmp protocol blocking process from the lan1 virtual machine to the virtual machine firewall is shown. Where in the process is the icmp protocol that enters the firewall in DROP.

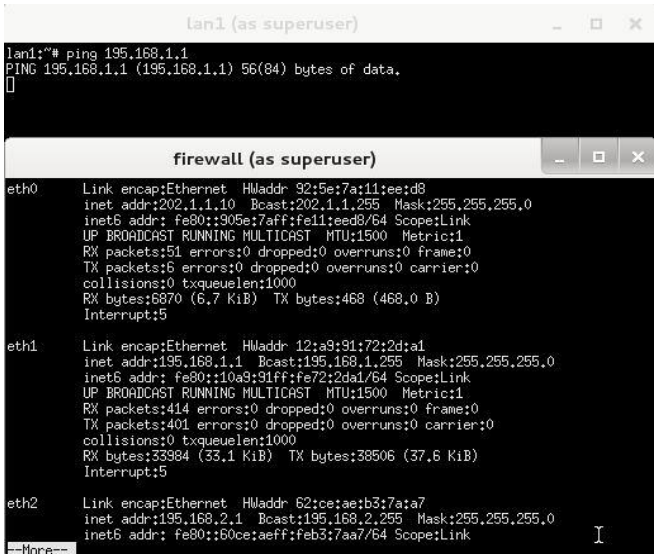

Fig . 9: DROP ICMP Protocol lan1 to Firewall

In Fig. 10, it shows the process of obtaining the icmp protocol access from the lan2 virtual machine to the virtual machine firewall. Where in the process is the icmp protocol that enters the firewall in ACCEPT, it shown with connected ping command.

Fig. 10: ACCEPT ICMP Protocol lan2 to Firewall

In Fig. 11 shows the process of obtaining ac-cess to port 22 SSH to remote access from the virtual machine lan2 to a virtual machine firewall. Where the virtual machine firewall opens port 22 SSH access for the virtual machine lan2.

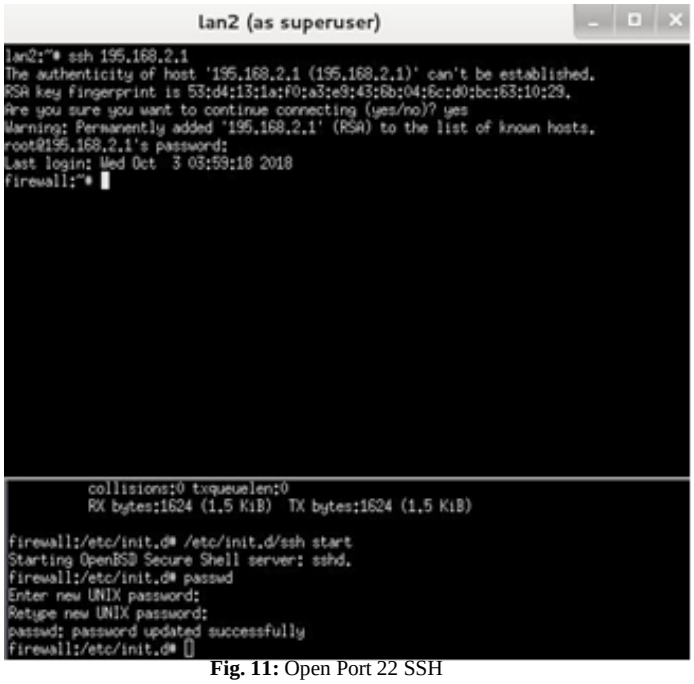

Fig. 11: Open Port 22 SSH

5.3. Analysis Testing Virtual Lab.

Analysis of virtual machine experiment in virtual lab. firewall iptables in table 3.

Table 3: Analysis Iptables Firewall

No	VM 1	VM2	Rule Iptables	Status
1	lan1	firewall	DROP ICMP Protocol	Running
2	lan2	firewall	ACCEPT ICMP Protocol	Running
3	lan2	firewall	Open Port 22 SSH	Running

6. Conclusions

From the results of the virtual lab test, it can be concluded that all the rules of the iptables firewall can be run in a virtual lab by using a netkit emulator. By creating a file service to write the rules of the firewall iptables in the firewall.sh file, it will be easier for teacher and student to update and correct firewall rules if needed. The firewalll.sh file can run directly by running the command to call the service on linux.

7. Contribution Author

The author's contribution in this paper is to design a linux network topology as a learning media for iptables linux firewall, which is implemented using netkit virtualization technology. Then the firewall iptables testing scenario is made by automating the writing of rules by creating service files, so that the teacher and students as users can experiment with the iptables firewall easily.

Acknowledgment

The authors would like to thank to the Indonesian Directorate General of the Higher Education (DIKTI) and State Polytechnic of Malang who have supported this research project.

References

[1] Ariyanto, Yuri. Watequlis S., Yan, Harijanto, Budi. Performance Analysis of Network Emulator Based On The Use Of Resources In Virtual Laboratory. Conference Eecsi 2017 - Yogyakarta, Indonesia, 19-21 September 2017. Publisher: IEEE Doi:10.1109/Eecsi.2017.8239075.

[2] Paulo H. M. Gurgela*, Luiz H. C. Brancob, Ellen F. Barbosaa,Kalinka R. L. J. C. Brancoa. Development of a practical computer network course through Netkit virtualization tool. 2013 International Conference on Computational Science. Procedia Computer Science 18 (2013) 2583 – 2586. Publisher : Available online at www.sciencedirect.com.

[3] Samuel T. King, George W. Dunlap, Peter M. Chen, Operating System Support for Virtual Machines, Proceedings of the 2003 USENIX Technical Conference, pp 71-84, June 9–14, San Antonio, TX, USA, 2003.

[4] University of Roma Tre Computer Networks Research Group. Netkit. http://www.netkit.org accessed on 2 April 2018.

[5] Fermin Galan, David Fernandez, Javier Ruiz, Omar Walid, and Tomas de Miguel. Use of Virtualization Tools in Computer Network Laboratories. In Proc. 5th International Conference on Information Technology Based Higher Education and Training (ITHET 2004), pages 209-214, Jun 2004.

[6] Maurizio Pizzonia, Massimo Rimondini. Easy Emulation of Complex Networks on Inexpensive Hardware. Proc. 4th International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities (TRIDENTCOM 2008), Innsbruck, Mar 2008.

[7] Massimo Rimondini. Emulation of Computer Networks with Netkit. Technical Report RT-DIA-113-2007, Roma Tre University, Jan 2007.

[8] Hung Nguyen, Matthew Roughan, Simon Knight, Nick Falkner, Olaf Maennel, Randy Bush. How to Build Complex, Large-Scale Emulated Networks. Proc. 6th International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities (TRIDENTCOM 2010), Berlin, May 2010.

[9] Gregor N. Purdy. Linux Iptables Pocket Refences. O'Reilly. www.oreilly.com. 2004.

[10] [10] Qing-Xiu Wu. The Research and Application of Firewall based on Netfilter. 2012 International Conference on Solid State Devices and Materials Science. Procedia Computer Science 18 (2013) 2583 – 2586. Publisher : Available online at www.sciencedirect.com.

[11] Larry Peterson, Bruce Davie. Computer networks - A systems approach, 3rd Edition.Morgan Kauffman, 2004.