IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 20, Issue 6, Ver. I (Nov - Dec 2018), PP 40-51
www.iosrjournals.org

Performance of Network Security Issues in Open Source Servers Using iptables

Mohammad Jahangir Alam[1], Tanjia Chowdhury[2], Mohammed Hossain Ibrahim[3]
[1] *Assistant Professor, Department of Computer Science and Information Technology , Southern University Bangladesh, 739/A Mehedibag Road, Chittagong-4000, Bangladesh.*
[2]*Lecturer,Department of Computer Science and Information Technology , Southern University Bangladesh, 739/A Mehedibag Road, Chittagong-4000, Bangladesh.*
[3] *Assistant Manager, Link3 Technologies Limited, Chittagong-4000, Bangladesh*
Corresponding Author: Mohammad Jahangir Alam

Abstract: *In network system, there are several Open source servers such as DNS, Mail, FTP, Web and SIP. To maintain a secure network environment every server administrator follows an effective method for controlling and operating the different servers and protect them from several attacks because Security is the main issue for networking system. In this research, we established a secure environment for open source based servers using packet filtering firewall tools known as iptables. To protect the open source servers from external attack we have applied some rules at Linux iptables on the basis of respective port numbers, allowing and disallowing particular IP address or IP addresses with subnet for the important protocols like TCP, UDP, RTP and ICMP. After applying rules on protocols, we have used a popular simulation software or network protocol analyzer known as Wireshark for checking how the iptables rules are work and shows the changes before and after applying rules. We have also deployed the security rules for SIP server which are also be applicable for other servers like DNS, SMTP, POP3, FTP, web and proxy server according to their service port. After applying all the rules in different servers we have got better result regarding security performance and servers are more secured then previous.*

Keywords: *Open Source Servers; Iptables; SIP server; Wireshark; Server Security.*

Date of Submission: 29-10-2018 Date of acceptance: 12-11-2018

I. Introduction

Open source software means those software whose source code is available to all users. Users can modify it anytime. Linux is an open source software and source code is also modifiable. For any open source software, Network security plays an important role. Network security issues involves with authorized and unauthorized access. Security means protects networks from threats and unauthorized access. Linux supports different types of servers such as Apache server, SSH server. There are also several important servers such as web servers, file servers, mail servers, etc., are used in a network. In this paper we focused on administrative control. To ensure security for open source server a system administrator has to follow some rules such as change the default login user name (root) and port (22) for remote SSH login to the server, use packet filtering rules for SSH port (22) usually transport control protocol (TCP) packets based on the IP address at Linux iptables. As some open source servers provide web based login known as graphical user interface, so apply packet filtering rules for HTTPS and TCP packets with port 443. To prevent internet control message protocol (ICMP) flood attack, administrator can allow IP addresses with subnet for ICMP echo ping and trace route request to the server and block rest of the IP address.

This research focused on the performance of server security in Linux environment. To implement server security, we have to understand the role of server monitoring tools and Linux inbuilt log management and maintaining the security of a servers. Monitoring the Linux kernel log files which usually at /var directory is very important because here administrator can see all the remote Secure Shell (SSH) login, web based login to the servers with the host IP address list. Data log also gives the list of IP addresses and port number which is more important to see the unauthorized attack. Here we used Wireshark simulation which showed the differences between before and after applying packet filtering rules to the Linux kernel. Wireshark is a free and open-source packet analyzer. It runs on various computing platforms including Windows, OS X, Linux, and UNIX. Wireshark is very similar to tcpdump (built in linux OS) but has a graphical front-end, plus some integrated sorting and filtering features.

II. Literature Review

In this section we have discussed previous related works for network security. Many researchers have worked with this topic. Ali Mohammed et al. was proposed a network secure environment for Linux. From this implementation and research of enhancing network security, we found that; security is not only limited in choosing a secured operating system or secured server configurations, but also related to both physical and application security configured in the network [2]. Another work by Kinjal Shah et al. focused on SIP server security. Computer security is categorized as three sections which is known as controls. The three controls are physical, Technical and administrative control [3]. Ashvini et al. was proposed a security in Linux OS. The security level of Linux depend on the system level and application level. To ensure security of Linux need to audit requirements. In Linux there are various configurations files which consists of security attributes. This paper encompasses on audit requirements and configuration files. The main objective of this paper is account policy and procedures. Another similar work by Muhammad Yeasir Arafat et.al was studied on Security Issue in Open Source SIP Server. The researchers focused on Open Source SIP Server security. In this research, authors have described prevention and technique of the SIP server from unauthorized access and attacks[4]. Sowgandh Sunil Gadi was described in details the construction of secure kernel for Anonymous FTP server, Web server, Mail server and a File server[5]. Other related works include, Amit K Nepal was proposed about "Linux Server & Hardening Security". In this paper author concentrated on the basic hardening of a Linux System, so that it may not be an easy target to prevent attacks. This document presents the basic security to secure the Linux Server as well as some most popular application services that commonly run on a Linux Server[6]. The work done in our research and the work done in above related works follow the same context of network or server security issues but the work that we have covered is to implement a security in open source servers. A popular simulation software or network protocol analyzer known as Wireshark is used to check how the iptables rules worked that applied for protocols and showed the changes before and after applying rules.

III. Overview of Common Attack in Open Source Servers

In computer networking system, an attack defines any attempt to destroy, expose, alter, disable, steal or gain unauthorized access to or make unauthorized use of an asset. In network system several common attack is happened which are showed in below Table 1.

Table I: Common Attack in Open Source Servers

Types of Attack	Description
Denial-of-service attack (DoS attack)	This cyber-attack occurs when the perpetrator try to disrupt a network resource or host temporarily to its intended user by superfluous requests.
TCP SYN Flood attacks	Disrupts TCP Three-way handshake procedures.
UDP Flood Attacks	By sending large amount of UDP packets to a target system, it disrupts network. It also spoofed the IP address of the UDP packet.
ICMP Attacks	This attack sends large amount of ICMP packets to the intended users and spoofed IP address. It also decreases bandwidth of the victim's network connection.
Memory Depletion Attacks	The client and server interaction happened through sending and receiving acknowledgement. In the mean time memory maintains the state information but the attacker deplete the memory and fragments the packets.
Bandwidth depletion Attacks	This attack doesn't consume resources of the physical server but it disrupts the capability of the hyperlink connecting the network to the server.

IV. Result and Analysis

4.1 Basic Security for the Linux Server

For security purpose, we apply the rules for remote Secure Shell (SSH) login to the Linux kernel such as changing the default username, password and port number. Do not use the default username root and port 22 as the SSH port. To ensure the primary level security, a system administrator should change these default user name and port number of SSH form the Linux kernel just following the bellow steps: Add a New Username and Password: To add and give a strong password for a new user has to give the command:

[root@proxy ~]# adduser ibrahim
[root@proxy ~]# passwd ibrahim
Changing password for user ibrahim.
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.

Now to ensure more security for a server change the default SSH port 22. To do this administrator has to change the directory of the linux as given bellow:

[root@proxy ~]# cd /etc/ssh where, "etc" is an essential directory of Linux kernel which provides the host specific system configuration files and has to move to the SSH directory. Here administrator looking for a file

which name is sshd_config, inside this file administrator can add the login permission of the new user here it is "ibrahim" and also can change the SSH port 10 instead of default port 22 by using the following command-

```
[root@proxy ~]# ls -la | grep sshd
-rw-------  1 root root   3331 Oct 12 12:40 sshd_config
 [root@proxy ~]#vi sshd_config
# $OpenBSD: sshd_config,v 1.70 2004/12/23 23:11:00 djm Exp $
# This is the sshd server system-wide configuration file.  See
# sshd_config (5) for more information.
AllowUsers ibrahim
Port 10
[root@salimbrothers ~]#  /etc/init.d/sshd restart
Stopping sshd:                             [OK]
Starting sshd:                             [OK]
```

Now SSH remote login default username root is disabled and Linux kernel allow only user "ibrahim" which is created earlier for the remote SSH login and port 10 will be used instead of 22 with the hostname or IP address of the asterisk server. Then the user "ibrahim" can login to the "root" via below command with root password:
[link3@proxy ~]$ su - ; Password: [root@proxy ~]# . So this may help something administrator to provide multilevel password protection to his server and unauthorized remote sessions to his server from different hosts. To monitor the authorized sessions, administrator can check the log file of the Linux regularly. From the log file we could see hundreds of failed register attempts from the several IP addresses in internet. This log may supply vital information to fight hackers because it shows the scale of the issue, the accounts they are attempting to crack, nature of attack and also attacker IP address.

4.2 Log Monitoring
The useful variable directory "var" which contains the entire log file of Linux kernel by using following command:

Fig.1 To see secure log in Linux Kernel

Using the following command we can see the Directories in Linux kernel: **[root@masintimates ~]#tail –f / var / log / Messages**

```
[root@masintimates ~]# tail -f /var/log/messages
Mar 29 07:14:09 localhost xinetd[2808]: START: tftp pid=10106 from=184.105.139.118
Mar 29 07:14:09 localhost in.tftpd[10107]: RRQ from 184.105.139.118 filename a.pdf
Mar 29 07:14:09 localhost in.tftpd[10107]: sending NAK (1, File not found) to 184.105.139.118
Mar 29 07:29:09 localhost xinetd[2808]: EXIT: tftp status=0 pid=10106 duration=900(sec)
Mar 29 07:36:59 localhost xinetd[2808]: START: tftp pid=10233 from=64.39.99.88
Mar 29 07:36:59 localhost in.tftpd[10234]: RRQ from 64.39.99.88 filename kernel
Mar 29 07:36:59 localhost in.tftpd[10234]: sending NAK (1, File not found) to 64.39.99.88
Mar 29 07:36:59 localhost in.tftpd[10235]: RRQ from 64.39.99.88 filename kernel
Mar 29 07:36:59 localhost in.tftpd[10235]: sending NAK (1, File not found) to 64.39.99.88
Mar 29 07:51:59 localhost xinetd[2808]: EXIT: tftp status=0 pid=10233 duration=900(sec)
```

Fig.2 To see the server information log of Linux Kernel

The Linux kernel log also gives the all successful remote SSH login session with IP address or hostname, login time-date month, username and time of the active sessions. To view this administrator can run a single command that given below in Figure 3:

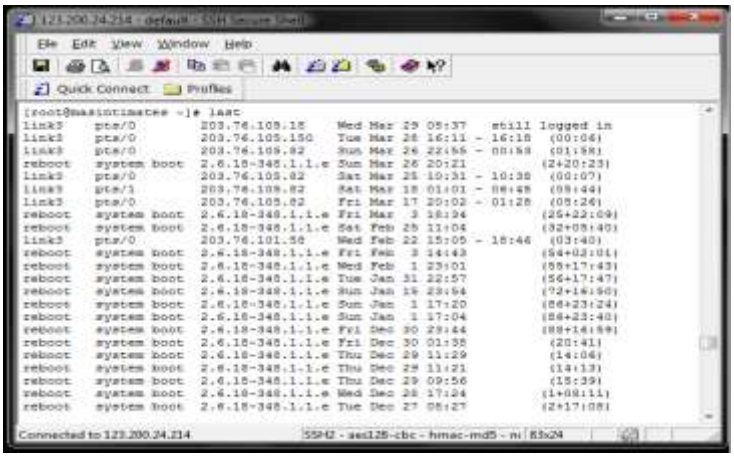

Fig.3 The list of Host IP addresses of SSH login to the SIP server

4.3. Apply Extreme Level Security for Open Source Server using Linux iPtables

iPtables which is a supply filtering firewall of Linux kernel and it provides the users to change and construct an individual freedom. Net filter is the fundamental concept of IPtables. It uses in the field of IP, TCP, UDP, and ICMP packet headers etc. By Applying rules at Linux iptables administrator can allow or prohibit a certain IP address or range of IP addresses with the foundation of port number for a certain asterisk server. To make more security for the asterisk server administrator can apply rules at Linux IPtables for remote SSH login, Web Browsing (https, http). Network mapping "nmap" in Linux is a critical command to see the open protocols with port number as below command output:

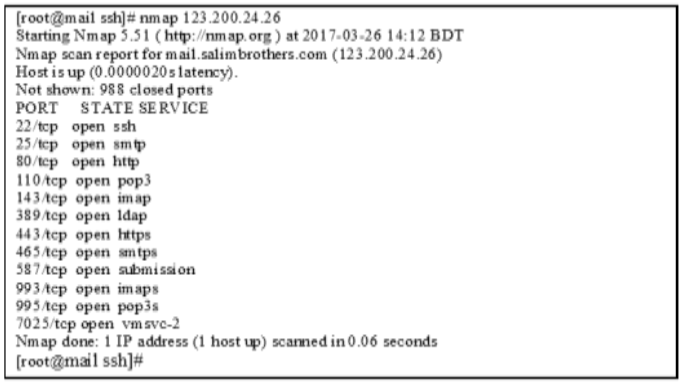

Fig.4 To see the Port/service status of Mail server

4.4 Iptables rules for remote SSH login to the SIP server:

Session Initiation Protocol (SIP), an application-layer control (signaling) protocol used for creating, modifying, and terminating sessions with a number of applicants. A SIP message includes a header by having an elective body. These messages are either requests or responses. It requests through the default port 5060 for allowing and cancelling IP addresses using IPtables rules. SIP entity received request and it performs the corresponding action and sends back an answer to the originator of the request. Responses are three digit status codes.

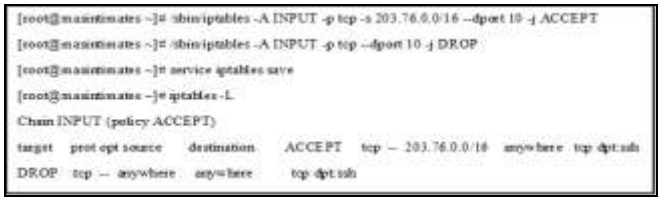

Fig. 5 Iptables rules for remote SSH login to the SIP server

Without applying the rules at Linux iptables for SIP default port 5060, it is seen that a user agent with IP address 203.76.105.82 provides 2 register packet requests to the SIP server 123.200.24.214 according to the Figures 7 and 8 at wireshark then the SIP server 123.200.24.214 provides the OPTIONS and successful response code 200 OK as an acknowledgement message to the user agent 203.76.105.82 as given as below Figures 9, 10 at wireshark.

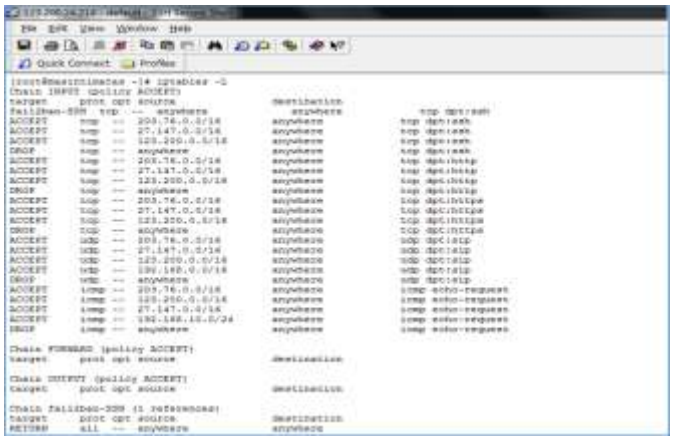

Fig.6 To display all rules in iptables of Linux kernel

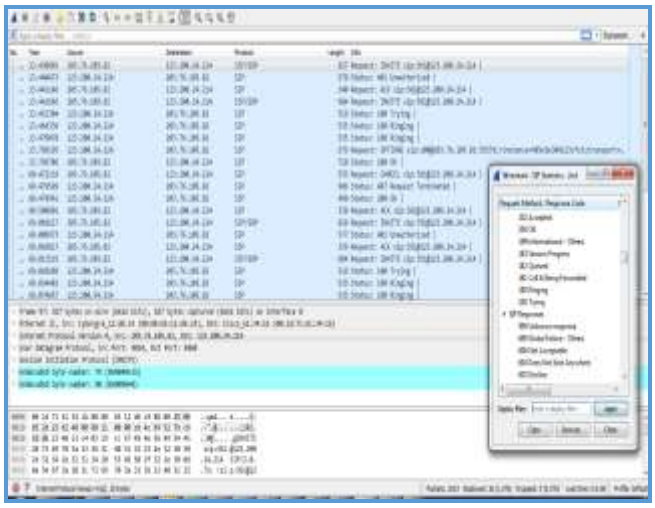

Fig.7 SIP register request packets flow form IP 203.76.105.82 to SIP server 123.200.24.214

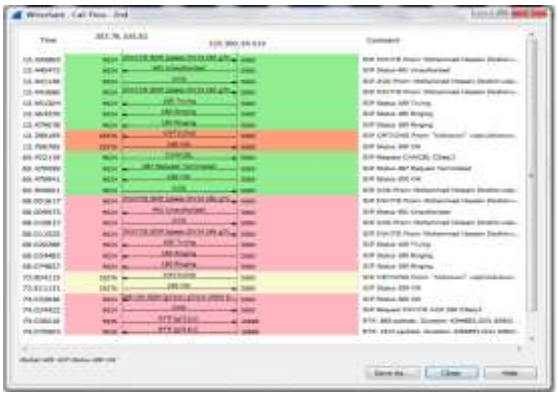

Fig.8 Flow graph of SIP register request from the host 203.76.105.82 to SIP server 123.200.24.214

Now apply the rules at Linux kernel iptables to allow the IP block 203.76.0.0./16 and cancel rest of the IP addresses for SIP Register request through the default port 5060 for the SIP server with IP address 123.200.24.214 by using bellow commands.

```
[root@masintimates ~]# /sbin/iptables -A INPUT -p udp -s 203.76.0.0/16  --dport 5060 -j ACCEPT
[root@masintimates ~]# /sbin/iptables -A INPUT -p udp --dport 5060 -j DROP
[root@masintimates ~]# service iptables save
Saving firewall rules to /etc/sysconfig/iptables:          [  OK  ]
[root@masintimates ~]# iptables -L Chain INPUT (policy ACCEPT)
target   port    opt source      destination
ACCEPT udp --  203.76.0.0/16  anywhere udp dpt:sip
DROP      udp  --  anywhere    anywhere  udp dpt:sip
```

Now send register request form the user agent with IP address 192.168.43.192 which is not at allowed IP block therefore Linux kernel dropped the SIP register request through the agent 192.168.43.192 send 15 packets of register request to the SIP server 123.200.24.214 according to the wireshark flow graph in Figure 9.

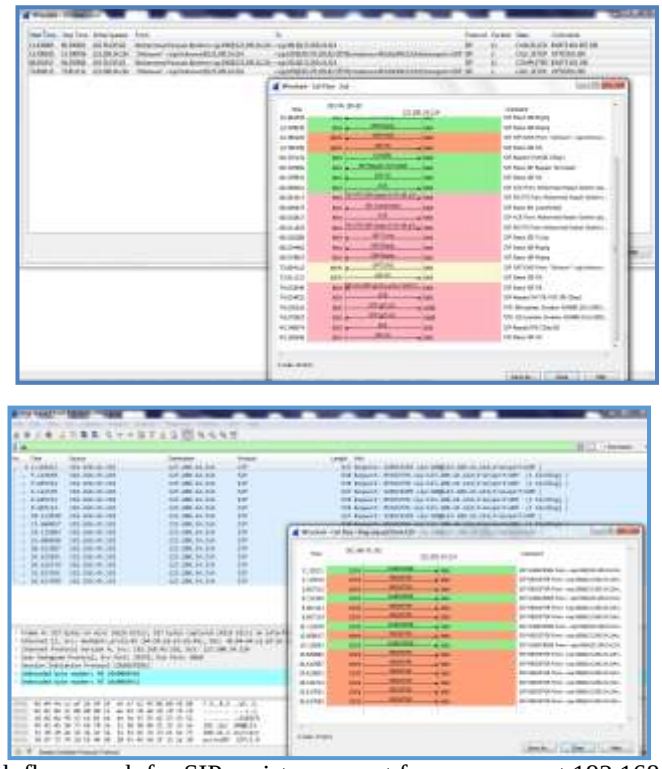

Fig. 9 Wireshark flow graph for SIP register request from user agent 192.168.43.192 to SIP server 123.200.24.214

4.5 Prevent Internet Control Message Protocol (ICMP) Flood Attack

ICMP works at the network layer and ICMP flood attack which enables users to send an echo packet to a remote host to check whether it's alive or not. Without applying rules at Linux iptables for ICMP ping request at the server, it allows input ping request packet from different host and then transmit an echo or reply packet to that host as an acknowledgement as like as Figure 10. Here a host 203.76.105.82 which OS is windows-7 sends ping request to the mail server which OS is Red Hat Linux (centos-6.8) with ip address 123.200.24.26 Since still Linux sever does not apply iptables rules for ICMP, host 203.76.105.82 gets continuous TTL response as wireshark shows the ping request from 203.76.105.82 and reply request from 123.200.24.26. Therefore Packet/Tick graph at wireshark gives the continuous line as like as below Fig. 11.

Fig.10 Wireshark view for ping request from 203.76.105.18 to 123.200.24.214 and packet/tick graph at wireshark

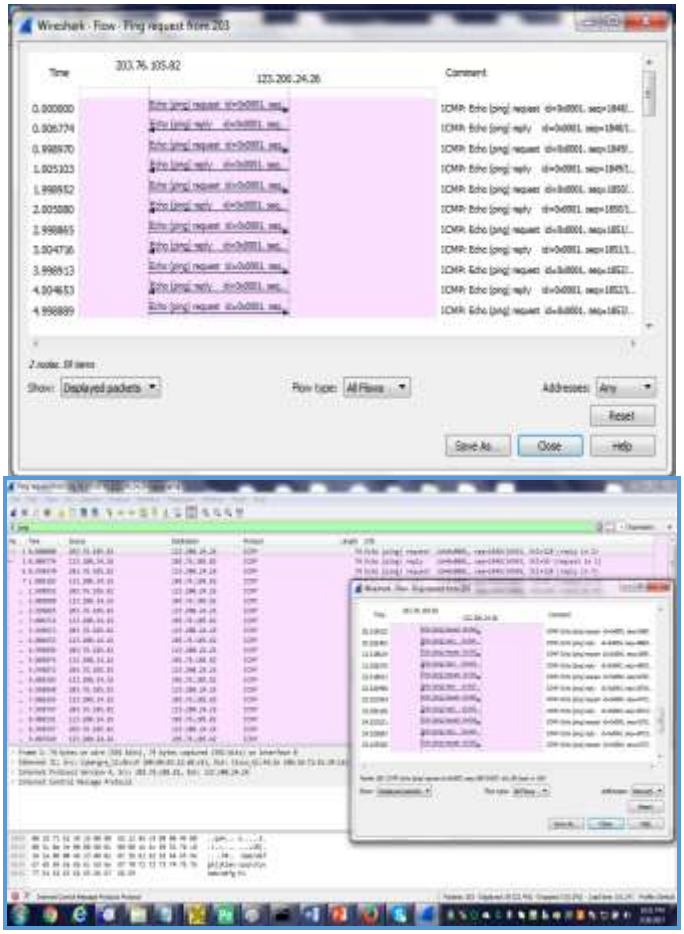

Fig.11 Wireshark flow graph for ping request from 203.76.105.82 to 123.200.24.26

Without ICMP rules at Linux firewall the trace route command by host 203.76.105.82 gives the complete path to reach the mail server 123.200.24.26 as given below Fig. 12.

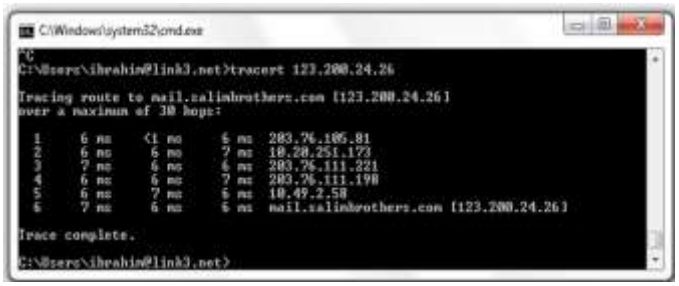

Fig.12 Trace route path from the host 203.76.105.82 to mail server 123.200.24.26

Without ICMP rules at Linux firewall the trace route command by host 203.76.105.18 gives the complete path to reach the SIP server 123.200.24.214 as given bellow Fig.13.

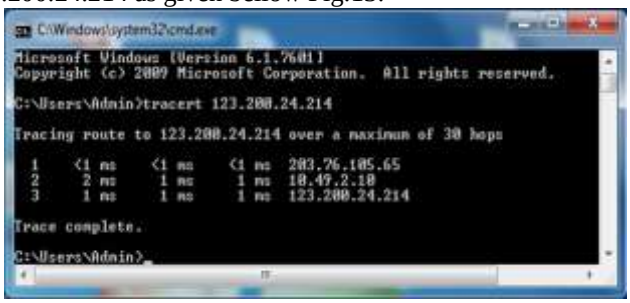

Fig.13 Trace route path from the host 203.76.105.18 to SIP server 123.200.24.214

Now apply the rules for ICMP at iptables of Linux kernel by using below commands, here input icmp echo-request of the server with IP 123.200.24.214 is allowed only for IP block 203.76.105.0/26 and rest of the IP addresses of the world will be dropped when they send the icmp ping request to this asterisk server with IP address 123.200.24.214 by the Linux kernel.

[root@masintimates ~]# iptables -A INPUT -s 203.76.105.0/26 -p icmp --icmp-type echo-request -j ACCEPT
[root@masintimates ~]# iptables -A INPUT -p icmp --icmp-type echo-request -j DROP
[root@masintimates ~]# service iptables save
Saving firewall rules to /etc/sysconfig/iptables: [OK]

After applied above rules for ICMP at iptables of Linux kernel following figure 14 shows the allowed only for IP block 203.76.105.0/26 and rest of the IP addresses of the world will be dropped:

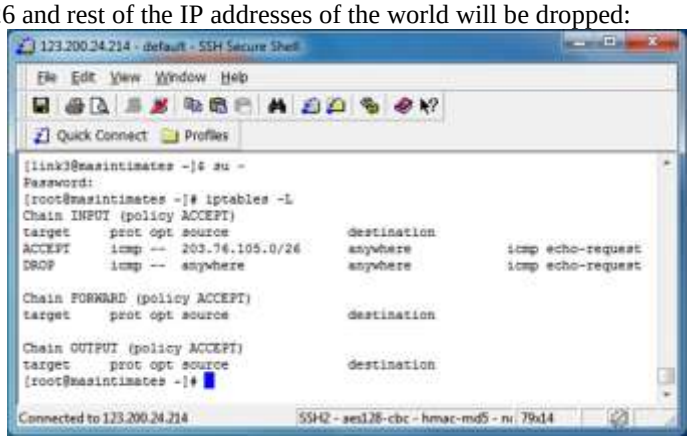

Fig. 14 To view applied rules for ICMP at iptables of Linux kernel

Now the ping request sender 27.147.249.54 will get request time out as an acknowledgment message from the host 123.200.24.214 because Linux kernel will dropped the ping request packets as the kernel allow only the IP block 203.76.105.0/26 and there will be no reply request packet to the host 27.147.249.54 from 123.200.24.214 as wireshark flow graph shows at Fig.15.

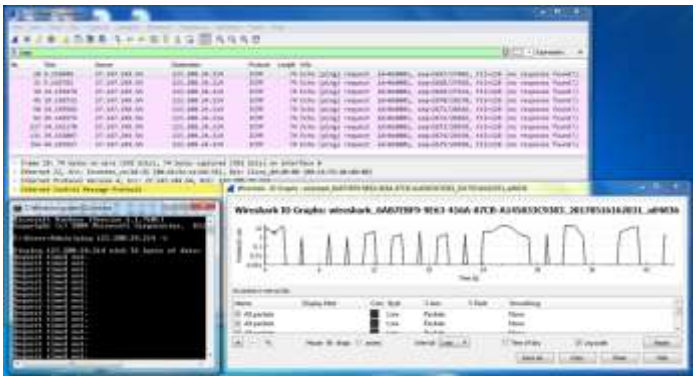

Fig.15 Wireshark view for the ping request from host 27.147.249.54 to Linux server with IP address 123.200.24.214 and packet /tick graph at wireshark.

Now the trace route command by host 27.147.249.54 shows the request time out response and failed to reach the SIP server 123.200.24.214 as given in Figure 16 due to applied rules for ICMP at Linux iptables.

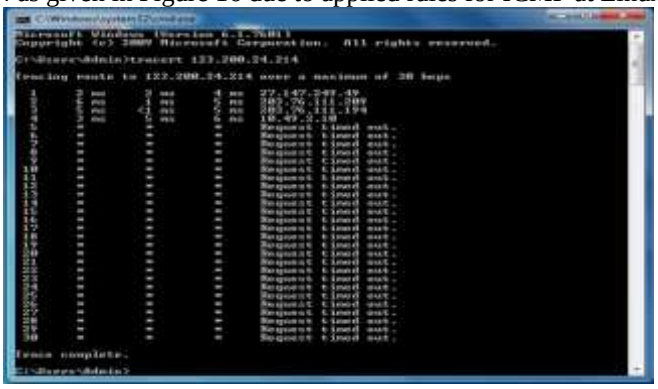

Fig. 16 Trace route path from the host 27.147.249.54 to SIP server 123.200.24.214 after applying ICMP firewall rule

Linux iptables show the accepted and dropped packets according to apply rules as like as bellow command output in Figure 17.

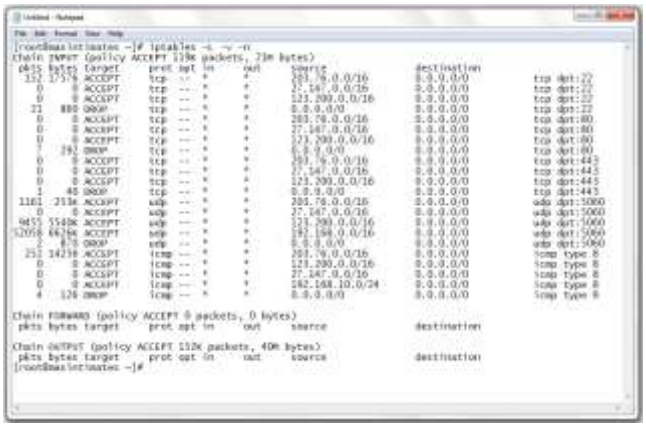

Fig. 17 Accepted and dropped packets list according to rules applied at Linux firewall/iptables

Administrator can delete existing rules at Iptables through following steps as given below:
Suppose user wants to delete the rules number 2, then follow steps given bellow:

[root@masintimates ~]#iptables -D INPUT 2
[root@masintimates ~]#servcie iptables save
Command to restart and Flush Linux iptabes
[root@masintimates ~]#service iptables restart
[root@masintimates ~]#iptables –F

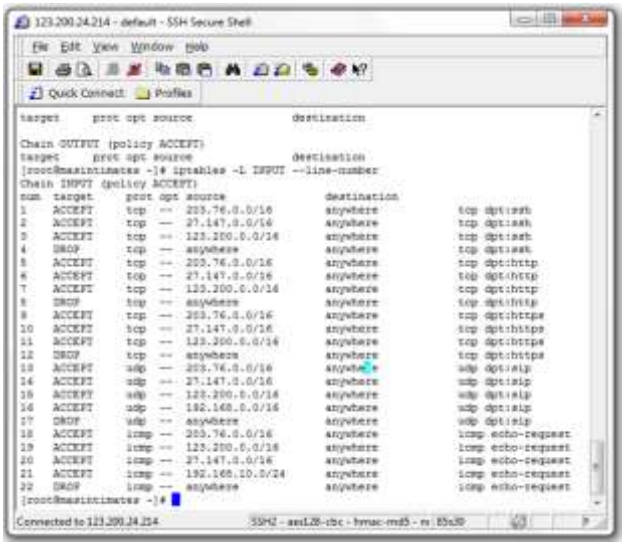

Fig.18 Command to see the iptables rules with the line number applied rules at Linux firewall/iptables

4.6 Extensions Security Using Fail2ban in Open Source Servers

To prevent SIP flood by fail2ban need to create a new filter configuration for Asterisk:

touch /etc/fail2ban/filter.d/asterisk.conf. After using 3 times wrong authentication attacker IP address will automatically block in SIP server. The block IP address are given in Fig. 19.

[root@mail ~]# cat /var/log/messages

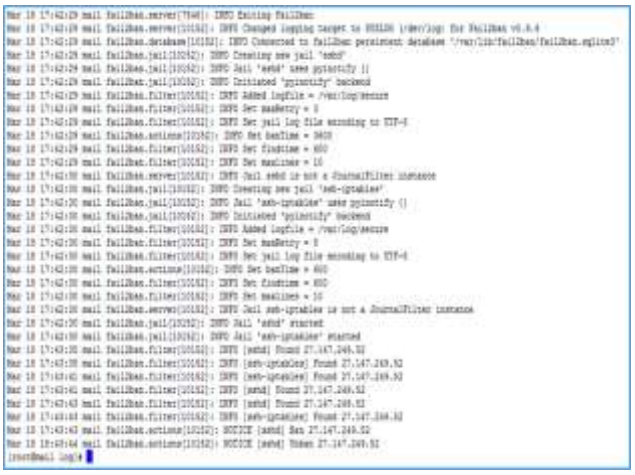

Fig.19 Attacker blocked IP address

We have made some failed login attempts from one of our server to the server where fail2ban installed and it works. You see the banned IP address of my server. Message from syslogd@tecmint at Nov 23 13:57:53... fail2ban.actions: WARNING [ssh-iptables] Ban 27.147.249.52 Verifying Fail2Ban iptables rules in following Fig 20.

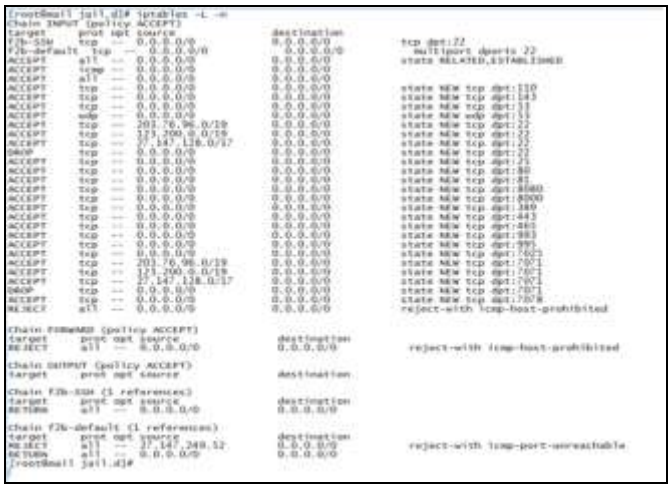

Fig.20 Verifying Fail2Ban iptables rules

4.7 Performance analysis graph

According to secure log analysis of servers we have found that attackers tried to attempt login to server thousands of time in 10 days but they have failed to login to servers and attackers could not login to servers even for a single time. As a result our applied rules in different server have protected and secured the servers. Following graph shows the failed attempt in servers in 10 days report:

Table II Performance Analysis Graph

Server's Name	Total Failed Attempt
Mail Server1(27.147.240.230)	11029
Mail Server2(123.200.24.26)	97245
Mail Server3(27.147.249.50)	97099
Proxy Server(123.200.31.70)	100508
SIP Server(123.200.24.214)	476401

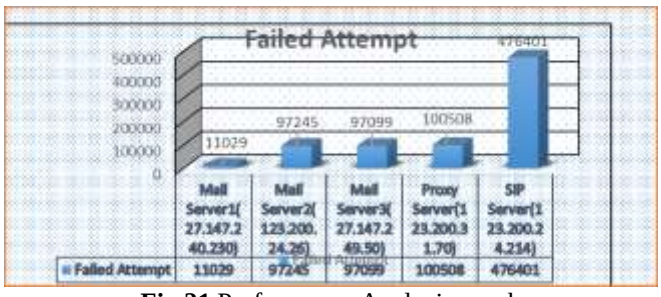

Fig.21 Performance Analysis graph

V. Conclusion

In this research, we have implemented different iptables/firewall rules to protect open source servers from different types of attacks. By changing some default ports and implementing firewalls/iptables rules, we can protect our servers easily. We have analyzed how Linux iptables rule work with wireshark simulation screening the differences of filtering packets before and after applying rules. Here we ensured security for port number and some useful protocols such as SSH, ICMP, HTTPS, SIP, SMTP, POP3 by filtering TCP and UDP packets with allowing a particular IP addresses with subnet mask and disallowing rest of the addresses of the world to prevent unauthorized attack to the server. This paper demonstrates how to choose strong password pattern that ensured more security to the servers. In this paper we also implemented a real time notification and monitoring of any intrusions in the server so that a system administrator may respond quickly which can be helpful for administrator to protect any kind of attacks. After implementing the different rules maximum attacks can be prevented and servers are found more secure than previous implemented rules.

References

[1]. William Stallings, *Cryptography and Network Security Principles and Practices*,(4th ed, Prentice Hall,2005).

[2]. Ali Mohammed, Sachin Sama and Majeed Mohammed, *Enhancing Network Security in Linux Environment*, School of Information Science, Computer and Electrical Engineering, *Halmstad University, Sweden, February* 2012.

[3]. Kinjal Shah, Satya Prakash Ghrera, Alok Thaker, A Novel Approach For Security Issues In VOIP Networks In Virtualization With IVR. *International Journal of Distributed and Parallel Systems (IJDPS),3(3),* May 2012.

[4]. Muhammad Arafat, Muhammad Alam, Feroz Ahmed, Study on Security Issue in Open Source SIP Server , *Modern Applied Science, 8(2),* 2014.

[5]. Sowgandh Sunil Gadi, *Security hardened kernels for linux servers*, *Wright State University*, 2004.

[6]. Amit K Nepal, Linux Server & Hardening Security, August 2013.

[7]. Hongjuan Li ,Yuqing Lan, "A Design of Trusted Operating System Based on Linux", IEEE International conference on Electrical and control Engineering, pp 4598 – 4601,Nov.2010.

[8]. Richard Sharpe, Ed Warnicke "Wireshark", [Online].Available: http://www.wireshark.org/docs/wsug_html/#ChapterIntroduction

[9]. Bhisham Sharma , Karan Bajaj, Packet Filtering using IP Tables in Linux, *IJCSI International Journal of Computer Science Issues,8(4),* July 2011.

[10]. Ali Mohammed, Sachin Sama and Majeed Mohammed, Enhancing Network Security in Linux Environment, *Technical Report, IDE-1202*, February 2012.

[11]. Ehlert, S., Geneiatakis, D., & Magedanz, T, Survey of network security systems to counter SIP-based denial-of-service attacks, *Computers & Security,vol.29(2),*March, 2010

[12]. Rani, D. D, Krishna, T. S, Dayanandam, G, & Rao, T. V , TCP Syn Flood Attack Detection And Prevention, *International Journal of Computer Trends and Technology (IJCTT), 4(10),* Oct. 2013.

Mohammad Jahangir Alam. " Performance of Network Security Issues in Open Source Servers Using iptables " IOSR Journal of Computer Engineering (IOSR-JCE) 20.6 (2018): 40-51.