IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Issue-5, May 2019 E-ISSN: 2347-2693

Network Resource Management using Policy based Inward Traffic

Ishwar Dayal Singh[1*], O. P. Gupta[2]

[1]Department of Electrical Engineering and Information Technology, Punjab Agricultural University, Ludhiana, Punjab, India
[2]Information Technology section, College of Agriculture, Punjab Agricultural University, Ludhiana, Punjab, India

DOI: https://doi.org/10.26438/ijcse/v7i5.12001203 | Available online at: www.ijcseonline.org

Accepted: 23/May/2019, Published: 31/May/2019

***Abstract*—** The security and management of network has become a major issue in the arena of Internet. The attacker can access different types of data i.e. personal data, bank account data, and unauthorized use of system resources in campus network. Various policies and procedures have been developed to secure the network communication over the internet by employing firewalls, encryption, and virtual private networks. On the bases of security requirements, the firewall rules are created to monitor the incoming traffic. Packet filtering technique has become a regular and inexpensive approach to secure the transfer of data over the internet and is used as a first line of defense against attacks. IPTABLES is used to create, maintain and monitor packet filter rules in the Linux operating system. Strong filtering techniques in IPTABLES can be used to make a network robust in nature for securing data transfer or prevent it from attacks. In this paper, study is done, not only to safe guard the network Distributed Denial of Service (DDOS) attacks but also management the network bandwidth. The proposed policy script based on the size and count of packet, blocks the attacker for a period of time. With the use of this policy, it observed that 33.8% bandwidth is always available to genuine users of the IT services.

***Keywords*—** Network Security, Packets, IPTABLES, Policy Script, Packet Count

I. INTRODUCTION

The Internet is becoming very essential part of human life. Everyday people interact with the world through it. Banking, shopping, business, transactions, E-mailing and many more applications are carried out over the internet. With the growth of services provided by the internet, a lot of problems like malicious software, viruses and hacking activities are associated with it that can impact security and privacy of human life. For large companies, security breaches means loss of millions of dollars that is why computer experts have worked on developing security models that protect the systems from attacks that threaten the confidentiality, integrity and availability of the information.

Network Security being a prime task must be considered thoughtfully when a network is designed. The network traffic flow is controlled effectively by employing a firewall policy that decides the filtering procedure. The packet will be accepted or rejected for accessing the network by checking against defined rules.

As the valuable information resides on network, many network defenses have been designed. In this paper, policy script is developed in Linux that counts the number of incoming packets. When the packets received by network becomes greater than the prescribed limit then the policy script starts dropping those packets to load balance the bandwidth and maintaining the smooth flow of traffic in the network.

Section I contains the introduction of Network Security, Section II contains the brief overview of Operating System and IPTABLES. Section III explain the methodology of flow of packets through the IPTABLES with the flow chart, Section IV contains the implementation of the experiment and the results are obtained in the form of tables, Section V describes the results and discussion, Section VI contains the conclusion of the research paper.

II. OPERATING SYSTEM AND IPTABLES OVERVIEW

Linus Benedict Torvalds developed an open source operating system named Linux in August 25, 1991. The developers can access all the source codes, integrate new functions and also eliminate programming bugs quickly.

In this study, CentOS is used as an operating system. It is a stable, predictable, manageable and reproducible platform. A Firewall is hardware, software or a combination of both that monitors and filters traffic packets. It performs the two basic security functions such as Packet Filtering based on rules of policy script as well as at the same time protects the host network from attackers. It prevents unauthorized access to network by maintaining the security of computers connected in LAN or WAN by using various types of signatures and host conditions. The Firewall performs the

various tasks such as defending the resources, managing and controlling network traffic, records and reports on events etc.

I.I IPTABLES

IPTABLES is christened for controlling the IP addresses available on the internet. It is a Linux command line firewall that allows system administrators to manage incoming and outgoing traffic via a set of configurable table rules. It is a file name as well as used as a command in Linux. It uses a set of tables which have chains that contain set of built-in or user defined rules. Three queues of IP tables for inward and outward traffic are used such as FILTER, NAT and MANGLE. The FILTER queue is responsible for packet filtering and has three built-in chains in which policy rules are placed. The NAT queue is responsible for network address translation and has two built-in chains. The MANGLE queue is responsible of the alteration of quality of service bits in TCP header.

Structure of IP Tables
IP tables commands have the following structure:
Iptables [-t <table-name>] <command> <chain-name> <parameter-1> <option-1>\ <parameter-n> <option-n>
<table-name>:- table on which the rule is applied.
<command>:- the action to perform, such as appending or deleting a rule.
<chain-name>:- the chain to edit, create or delete.
<parameter> *<option>* pairs:- Parameters and respective options specify how to process a packet that matches the rule.

The length and complexity of an IP tables command can be changed significantly based on its requirements.

III. METHODOLOGY

Packets flow through IP tables permit the user to create the firewall rules precisely. As shown in Fig. 1, the three main chains INPUT, OUTPUT and FORWARD process the security policy.
The flow of packets through IP tables adopts the following steps:-

1. The incoming packets enter the PREROUTING chain that is used to NAT the destination on packets before the application of any rules.
2. Based on destination IP address of the packet, a ROUTING decision is taken.
3. The packet entering the INPUT chain will be compared to the rules and then allowed to transverse the OUTPUT chain only if the rules are followed.
4. If packet is not directed to the INPUT chain, then it will be processed by the forward chain where rules are checked. If accepted, the packet is routed to correct interface.
5. The packet will be dropped if IP forwarding is not enabled.
6. The outgoing packet is processed by the POSTROUTING chain, used to NAT the source address on packets after rules are applied.

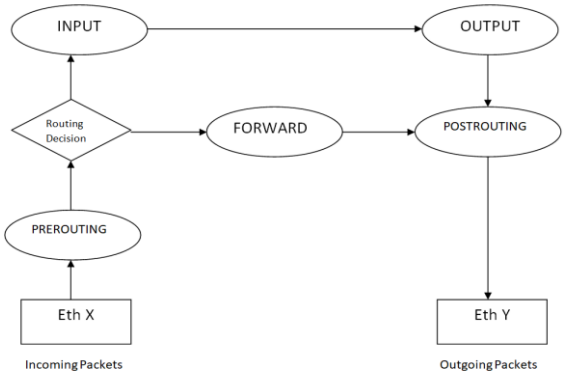

Figure 1. Packet Flow

IV. IMPLEMENTING DETAILS AND EXPERIMENTAL RESULTS

The experiment has been performed to monitor the inward traffic in the network based on the following:-
(1) Number of Packets of the inward traffic
(2) Packet Size

The following steps are performed while implementing proposed policy script:

Step 1: Traffic is analyzed using the *tcpdump* command in Linux. It helped in creating the logs of network traffic for the used interfaces.

Step 2: The Bash Scripting language is used for scripting in the IP Tables. Number of packets entering the network can be found using *ifconfig* command.

Step 3: The proposed script contains the limit of 60,000 packets entering in network per hour.

Step 4: When the packets starts entering the network from a source, then they are counted by the firewall using the proposed policy script. If the packet count exceeds the limit then the packets starts dropping in the network and the firewall blocks the IP address of that source.

The information flow in IPTABLES is shown in Figure 2.

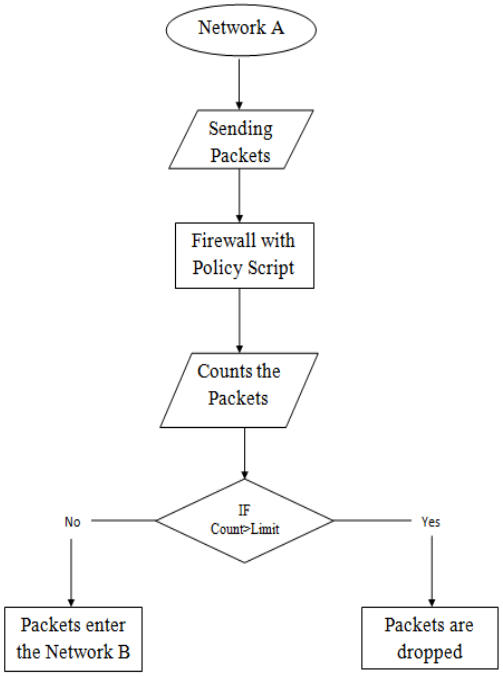

Figure 2. Logical Information Flow in IP Tables

There are two tables show the inward traffic per hour without and with blocking.

Table 1: Inward Traffic per hour without blocking

Hours	Packets	Bandwidth (%)
00	10554	17.6
01	13654	22.7
02	22023	36.7
03	38589	64.3
04	43246	72.1
05	55331	92.2
06	60000	100
07	60000	100
08	60000	100
09	60000	100
10	60000	100
11	60000	100
12	60000	100

Table 2: Inward Traffic per hour with blocking

Hours	Packets	Bandwidth (%)
00	10554	17.6
01	13654	22.7
02	22023	36.7
03	38589	64.3
04	43246	72.1
05	55331	92.2
06	60000	100
07	---	0
08	---	0
09	19771	33
10	26289	43.8
11	30743	51.2
12	39892	66.4

The following graph shows the dropping of packets when count exceeds the limit.

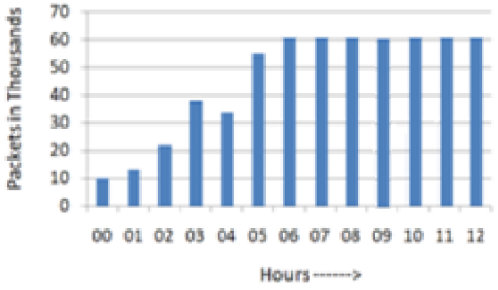

Figure 3. Inward Traffic without Blocking

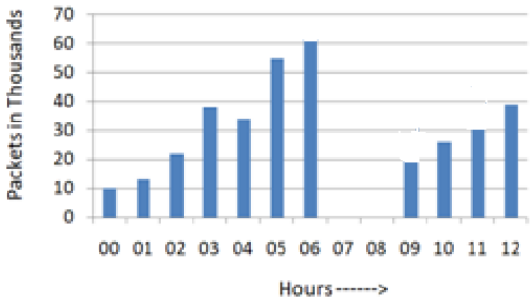

Figure 4. Inward Traffic with Blocking

The inward packets were monitored at 00 hour from a particular source. The traces of inward packets were varying with time. Initially 17.6 % of bandwidth of network was consumed by a particular source and 82.4 % is available to other sources. It slowly increases its consumption to 92.2 % which means only 7.8 % bandwidth was available for other sources. If that source was not blocked then, from Table 1, at 06 hour the source was sending 60,000 packets and reaches the maximum limit allowed. The source keeps sending 60,000 packets for remaining hours and completely consumed bandwidth and the other sources were unable to access the network and it completely choked it. But if that source was blocked, from Table 2, after sending 60,000 packets at 06 hours the proposed script started dropping the packets and the IP address was blocked for next two hours to free up the network resources. After two hours, the network

started receiving packets normally from that source. The network topology used is depicted in the figure 5.

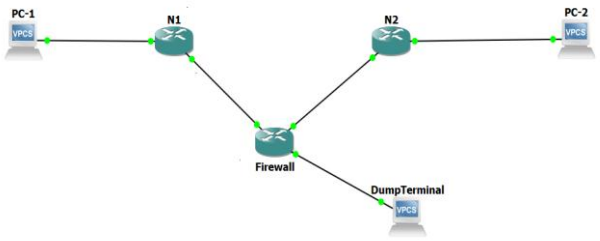

Figure 5. Network Topology

V. RESULTS AND DISCUSSION

It is analyzed that if that source was not blocked then the bandwidth consumption would be 83.8%. But if it was blocked for two hours then the bandwidth consumption would be 50%. By comparing the two values, it is concluded that 33.8% bandwidth consumption was protected from usage and become available to other sources and prevented the congestion in network flow. As per requirement, the blocking time can also be increased to more than two hours and more bandwidth will be available.

VI. CONCLUSION

In the present study, Firewall using IPTALES is acting as a gateway to the network and filtering based on packet count is implemented. A firewall is created with policy script that monitors and control the inward network traffic by counting the number of packets entering into the network. The script is implemented with the limit of 60,000 packets per hour from the particular source. When the packets cross the limit, the firewall starts dropping the packets and blocks the source IP, thereby reducing the congestion in the network and Distributed Denial of Service (DDoS) is reduced to such an extent that user will be able to get services. With the use of this policy, it observed that 33.8% bandwidth is always available to genuine users in the want of IT services.

REFERENCES

[1] S. Taluja, P. K. Verma, R. L. Dua, "*Network Security Using IP Firewalls*", International Journal of Advanced Research in Computer Science and Software Engineering, Vol. **2**, No. **8**, pp. 348-354, **2012**.

[2] K. Joshi, T. Kashiparekh, "*Implementing Firewall using IP Tables in Linux*", International Journal of Emerging Trends in Science and Technology, Vol. **3**, No. **3**, pp. 3634-3637, **2016**.

[3] S. Kadam, P. S. Tambabde, A. J. Jayant, "*Adaptive Packet Filtering Techniques for Linux Firewall*", International Journal of Advance Research, Ideas and Innovations in Technology, Vol. **3**, No. **1**, pp. 171-174, **2017**.

[4] B. Q. M. AL Musawi, "*Mitigating DoS/DDoS Attacks using IP Tables*", International Journal of Engineering and Technology, Vol. **12**, No. **3**, pp. 101-111, **2012**.

[5] B. Sharma, K. Bajaj, "*Packet filtering using IP Tables in Linux*". International Journal of Computer Science Issues, Vol. **8**, No. **4**, pp. 320-325, **2011**.

[6] N. D. Lal, B. Ghorbani, S. Vaghri, "*A Survey on the use of GNS3 for Virtualizing Computer Networks*", International Journal of Computer Science and Engineering, Vol. **5**, No. **1**, pp. 49-58, **2016**.

[7] W. Makasiranondh, S. P. Maj, D. Veal, "*Pedagogical evaluation of simulation tools usage in Network Technology Education*", World Transactions on Engineering and Technology Education, Vol. **8**, No. **3**, pp. 321-324, **2010**.

[8] A. M, M. Montero, D. R, Manzano, "*Design and Development of Hands-on Network Lab Experiments for Computer Science Engineers*", International Journal of Engineering Education, Vol. **33**, No. **2**, pp. 855-864, **2017**.

[9] A. Jalaparthi, A.S. Kumar, "*Monitoring and Analysis of Real time detection of traffic from twitter stream analysis*", International Journal of Scientific Research in Computer Science and Engineering, Vol. **4**, No. **3**, pp. 34-37**, 2016**.

[10] P. M, A. S. Thanamani, "*Real-time Packet Performances under Socket Application*", International Journal of Scientific Research in Network Security and Communication, Vol. **5**, No. **3**, pp. 84-89**, 2017.**

Authors Profile

Mr. Ishwar Dayal Singh is a student of Master of Technology in Computer Science and Engineering in the department of Electrical Engineering and Information Technology, Punjab Agricultural University, Ludhiana. He did his B.Tech in Computer Science and Engineering with distinction. He is having the passion for Network Management using open source softwares.

Dr. O.P. Gupta is an alumnus of Punjab Agricultural University Ludhiana, Thapar University Patiala and GNDU Amritsar, is working as Professor and Incharge, IT Section, COA, PAU, Ludhiana. He is bestowed with PAU Meritorious Teacher Award. His areas of interests include Parallel and Distributed Computing, Grid Computing for Bioinformatics, Network Testing and Network Management.