Research Article Impact Factor: 0.621 ISSN: 2319-507X
Manoj Namdeo Rathod, IJPRET, 2014; Volume 2 (8): 191-200 IJPRET

INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

INTERNET SECURITY USING IPTABLE

MANOJ NAMDEO RATHOD[1], K. B. MANWADE[2]

1. Department of Computer Science & Engineering, Shivaji University Kolhapur.
2. K. B. P., College of Engineering, Satara, India.

Accepted Date: 15/02/2014 ; Published Date: 01/04/2014

Abstract: Denial-of-Service (DoS) is a network security problem that poses a serious challenge to trustworthiness of services deployed on the servers. The aim of DoS attacks is to make services unavailable to legitimate users by flooding the victim with legitimate-like requests and current network architectures allow easy-to-launch, hard-to-stop DoS attacks. Threat of DoS attacks has become even more severe with DDoS (Distributed Denial-of-Service) attack .It is an attempt by malicious users to carry out DoS attack indirectly with the help of many compromised computers on the Internet. Attackers can compromise a huge number of computers by spreading a computer worm using vulnerabilities in popular operating systems. This exhausts the victim network of resources such as bandwidth, computing power, etc., the victim is unable to provide services to its legitimate clients and network performance is greatly deteriorated, moreover, with little or no advance warning, a DDoS attack can easily exhaust these resources within a short period of time. Service providers are under mounting pressure to prevent, monitor and mitigate DoS/DDoS attacks directed toward their customers and their infrastructure. Defending against those types of attacks is not a trivial job, mainly due to the use of IP spoofing and the destination-based routing of the Internet, though there are many proposed methods which aim to alleviate the problem like Firewalls, Traffic Volume Normalization, Intrusion Detection Systems, Ingress filtering, IP Traceback , SYN Proxy etc. *This work discusses about the efficient packet filtering technique using firewall to defend against DoS/DDoS attacks. Firewall scripts are written using command-line tool IP Tables in Linux to deny the suspicious traffic.*

Keywords: DoS attacks, DDoS attacks, iptables,, IP, Firewall

Corresponding Author: Mr. MANOJ NAMDEO RATHOD

Access Online On:

www.ijpret.com

How to Cite This Article:

Manoj Namdeo Rathod, IJPRET, 2014; Volume 2 (8): 191-200

PAPER-QR CODE

INTRODUCTION

Internet grows rapidly since it was created. Via the Internet infrastructure, hosts can not only share their files, but also complete tasks cooperatively by contributing their computing resources. Moreover, an end host can easily join the network and communicate with any other host by changing packets. These are the encouraging features of the Internet, openness and scalability. However, the attackers can also take advantage of this to launch attacks that are more powerful than those launched by a single machine. Denial-of Service Attack is one type of such attacks [1].A Denial of Service (DoS) attack is a type of attack focused on disrupting availability of service. Such an attack can take many shapes, ranging from an attack on the physical IT environment, to the overloading of network connection capacity, or through exploiting application weaknesses.[2] defined DoS as: "a group of otherwise authorized users of a specific service is said to deny service to another group of authorized users if the former group makes the specified service unavailable to the latter group for a period of time which exceeds the intended (and advertised) waiting time." Internet-facing and other networked infrastructure components are at risk of DoS.

II. Related work

The usage of internet is increasing rapidly. Via the Internet infrastructure, hosts can not only share their files, but also complete tasks cooperatively by contributing their computing resources. Moreover, an end host can easily join the network and communicate with any other host by changing packets. These are the encouraging features of the Internet, openness and scalability. However, the attackers can also take advantage of this to launch attacks that are more powerful than those launched by a single machine. Denial-of Service Attack /Distributed Denial-of Service Attack is one type of such attacks. A Denial of Service (DoS/DDoS) attack is a type of attack focused on disrupting availability of service. Such an attack can take many shapes, ranging from an attack on the physical IT environment to business environment. Therefore it is essential to detect and prevent DoS/DDoS attacks.

III. Types of DoS/DDoS Attacks

In DDOS attacks the attacker sends packets directly from his computer(s) to the victim's site but the source address of the packets may be forged. There are many tools available to allow this type of attack for a variety of protocols including ICMP, UDP and TCP. Some of the common DDoS attacks are discussed below.

III-A. UDP Flood Attack

In UDP Flood attack attacker sends large number of UDP packets to a victim system, due to which there is saturation of the network and the depletion of available bandwidth for legitimate service requests to the victim system. A UDP Flood attack is possible when an attacker sends a UDP packet to a random port on the victim system. When the victim system receives a UDP packet, it will determine what application is waiting on the destination port. When it realizes that there is no application that is waiting on the port, it will generate an ICMP packet of destination unreachable [5] to the forged source address. If enough UDP packets are delivered to ports of the victim, the system will go down. By the use of a DoS tool the source IP address of the attacking packets can be spoofed and this way the true identity of the secondary victims is prevented from exposure and the return packets from the victim system are not sent back to the zombies.

III-B. ICMP Flood Attack

ICMP Flood attacks exploit the Internet Control Message Protocol (ICMP), which enables users to send an echo packet to a remote host to check whether it's alive. More specifically during a DDoS ICMP flood attack the agents send large volumes of ICMP_ECHO_ REPLY packets (ping) to the victim. These packets request reply from the victim and this results in saturation of the bandwidth of the victim's network connection [4]. During an ICMP flood attack the source IP address may be spoofed.

III-C. SYN Flood Attack

In a SYN Flood attack, the victim is flooded with Half open connections. The client system begins by sending a SYN message to the server. The server then acknowledges the SYN message by sending SYN-ACK message to the client. The client then finishes establishing the connection by responding with an ACK message. The connection between the client and the server is then open, and the service-specific data can be exchanged between the client and the server. Fig. 1 shows the view of this message flow:

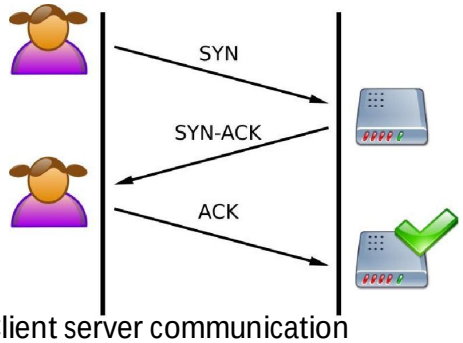

Fig.1 Client server communication

The potential for abuse arises at the point where the server system has sent an acknowledgment (SYN-ACK) back to client but has not yet received the ACK message. This is known as half-open connection. The server has built in its system memory a data structure describing all pending connections. This data structure is of finite size, and it can be made to overflow by intentionally creating too many partially-open connections. Creating half-open connections is easily accomplished with IP spoofing. The attacking system sends SYN messages to the victim server system; these appear to be legitimate but in fact reference a client system that is unable to respond to the SYN-ACK messages. This means that the final ACK message will never be sent to the victim server system.

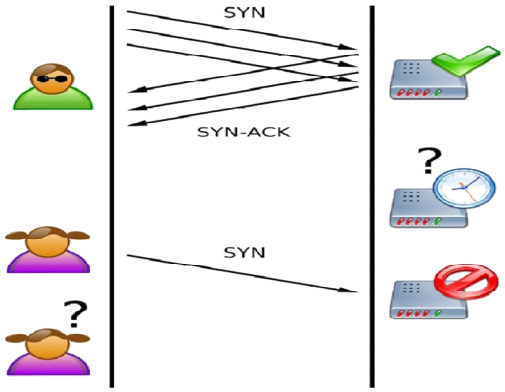

Fig.2 SYN Flood Attack

Fig.1 shows the scenario of half open connection. The half-open connections data structure on the victim server system will eventually fill; then the system will be unable to accept any new incoming connections until the table is emptied out [2].

III-D. Smurf Attack

In a "smurf" attack, the victim is flooded with Internet Control Message Protocol (ICMP) "echo-reply" packets. On IP networks, a packet can be directed to an individual machine or broadcast to an entire network. When a packet is sent to an IP broadcast address from a machine on the local network, that packet is delivered to all machines on that network. In the "smurf" attack [1], attackers are using ICMP echo request packets directed to IP broadcast addresses from remote locations to generate denial-of-service attacks. When the attackers create these packets, they do not use the IP address of their own machine as the source address. Instead, they create forged packets that contain the spoofed source address of the attacker's intended victim. The result is that when all the machines at the intermediary's site respond to the ICMP

echo requests, they send replies to the victim's machine. The victim is subjected to network congestion that could potentially make the network unusable.

III-E. Teardrop Attack

This type of denial of service attack exploits the way that the Internet Protocol (IP) requires a packet that is too large for the next router to handle be divided into fragments. The fragment packet identifies an offset to the beginning of the first packet that enables the entire packet to be reassembled by the receiving system. In the teardrop attack, the attacker's IP puts a confusing offset value in the second or later fragment. If the receiving operating system does not have a plan for this situation, it can cause the system to crash**.**

III-F. Land Attack

The attack involves sending a spoofed TCP SYN packet (connection initiation) with the target host's IP address to an open port as both source and destination. The reason a LAND attack works is because it causes the machine to reply to itself continuously. Land attacks have been found in services like Simple Network Management Protocol (SNMP) and Windows 88/tcp (Kerberos/global services) which were caused by design flaws where the devices accepted requests on the wire appearing to be from themselves and causing replies repeatedly [4].

IV. Experimental Setup and Measurements

The experimental setup was made by using Linux Ubuntu server version 12.10 and installed Wire shark, which is network protocol analyzer. Three modules attack will be studied in details as shown next:

IV-A. Module 1- SYN Flood Attack

The attack was made by flooding the victim's machine by running following Hping command from attackers:

 # hping3 --flood –S –p 80 192.168.0.1

Description:

--flood flag sends the packet as fast as possible

-S flag sets the SYN flag on in TCP mode

-p 80 sends the packet to port 80 on victim's machine (192.168.0.1) On victim machine, capturing and analyzing traffic using Wire shark show that victim machine (192.168.0.1) is responding to SYN packet by sending back packets with SYN, ACK flags set, but attacker's machine (192.168.0.3) is not participating three way handshake by sending back ACK, instead it is sending RST flag set packet thereby resulting in half open connection. When thousands of such connections are made in a few seconds, victim's resources will get exhausted in no time.

Fig.3 shows TCP flow graph using SYN flood attack

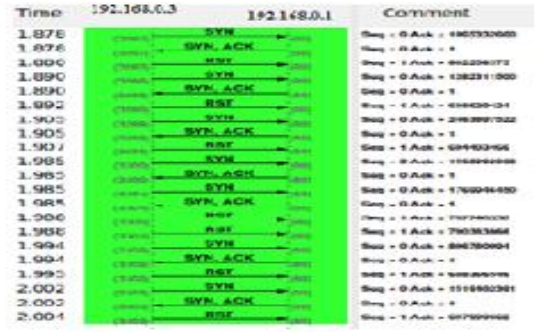

Fig.3

To defend against SYN Flood Attack, iptables script is writing as bellow.

 # iptables -N syn_flood

 # iptables -A INPUT -p tcp --syn -j syn_flood

 # iptables -A syn_flood -m limit --limit 1/s -- limit-burst 3 -j RETURN

 # iptables -A syn_flood -j DROP

Fig.4 shows TCP flow graph after

Applied the script and recapture packets using Wire shark to test working of the script.

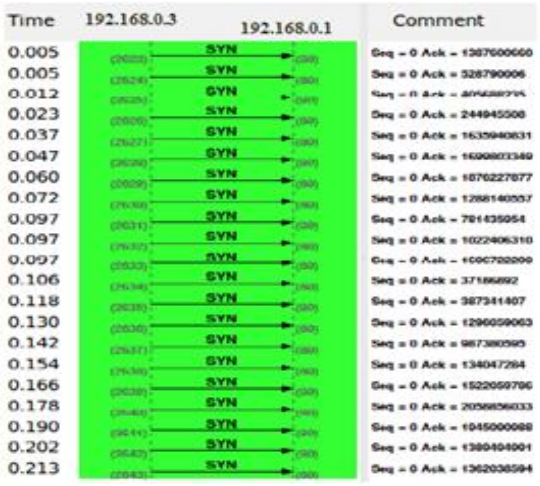

Fig.4

IV-B.Module 2- UDP Flood Attack

The attack was made by flooding the victim's machine by running following Hping command from attackers:

 # hping3 –p 80 –i u1000 --udp 192.168.0.1

Description:

-p 80 sends the packet to port 80 on victim's machine (192.168.0.1)

-i u1000 sets the interval between packets as 100 packets per second.
--udp flag sets the udp mode As seen in Fig.5 victim's machine
(192.168.0.1) is responding with ICMP port unreachable since there is no application running on attacker's machine which sent UDP packet. In this way all of the resources of victim's machine are consumed and legitimate requests will not be served as victim will be busy in serving attacker's invalid requests.

Fig.5
To defend against UDP Flood Attack, iptables script is writing as bellow
 # iptables -N udp_flood
 # iptables -A INPUT -p udp -j udp_flood
 # iptables -A udp_flood -m state –state NEW –m recent –update –seconds 1 –hit count 10 –j RETURN
 # iptables -A udp_flood -j DROP
Fig.6 shows UDP flow graph after
Applied the script and recapture packets using Wire shark to test working of the script

Fig.6

IV-C.Module 3-ICMP Flood Attack

The attack was made by Flooding the

Victim's machine by running following Hping command from attacker's:

 # hping3 –p 80 --flood --icmp 192.168.0.1

Description:

-p 80 sends the packet to port 80 on victim machine (192.168.0.1)

--flood flag sends the packet as fast as possible

--icmp flag sets the icmp mode

Fig.7 shows victim's machine

(192.168.1.2) under ICMP flood attack

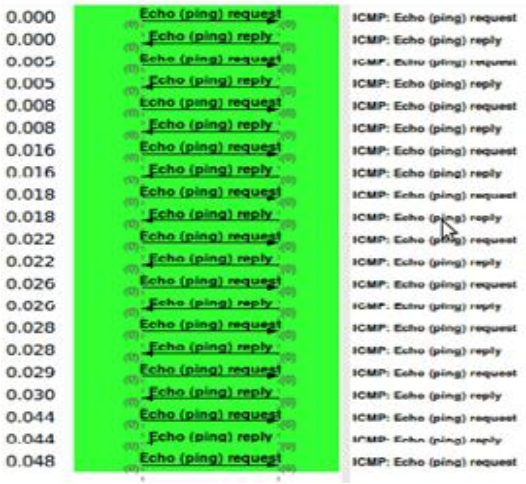

Fig.7

To defend against ICMP Flood Attack, iptables script is writing as bellow:

 # iptables -N icmp_flood

 # iptables -A INPUT -p icmp -j icmp_flood
 # iptables -A icmp_flood -m limit --limit 1/s - -limit-burst 3 -j RETURN
 # iptables -A icmp_flood -j DROP

Fig.8 attacker is sending ICMP Echo Request packets continuously but victim's machine is not responding by sending ICMP Echo Reply packets as all the packets are being dropped by the firewall according to the iptables rules.

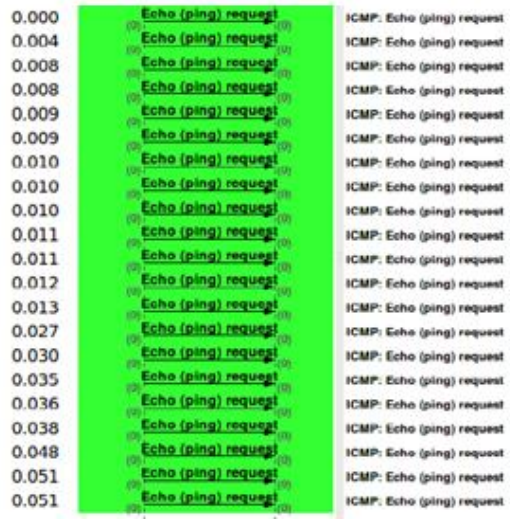

Fig.8

V. Conclusion

In this work, capability of firewall is explored to defend against this attack. To determine whether the network traffic is legitimate or not, a firewall relies on a set of rules it contains that are predefined by a network or system administrator. These rules tell the firewall whether to consider as legitimate and what to do with the network traffic coming from a certain source, going to a certain destination, or having a certain protocol type.

ACKNOWLDGMENT

Many people have shared their time and expertise to help me accomplish my goal. First, I would like to sincerely thank my guide, Mr. K.B. Manwade, Associate Professor, Computer Science & Engineering Department, Ashokrao Mane Group of Institutions, Vathar, Shivaji University, Kolhapur for his constant support and guidance. His instant responses to my countless inquiries have been invaluable and motivational. It was a great opportunity to work under his supervision.

Many thanks to Mr. G. A. Patil, Associate Professor and Head of Computer Science & Engineering Department, D. Y. Patil College of Engineering & Technology, Kolhapur, Shivaji

University, Kolhapur for his moral support and research environment he had facilitated for this work

REFERENCES

1. Saman Taghavi Zargar, James Joshi, David Tipper," *A Survey of defense Mechanism against Distributed Denial of Service (DDoS) Flooding Attack",* IEEE communication survey and tutorial, 2013.

2. Jerome Francois, Issam Aib, Raouf Boutaba," *FireCol: a Collaborative Protecting Network for the Detection of Flooding DDoS Attack",* IEEE Transaction on networking, December 2012

3. Khaled Salah, Khalid Elbadawi, Raouf Boutaba," *Performance Modeling and Analysis of Network Firewall"*, IEEE Transaction on network and service management, March 2012.

4. Udi Ben-Porat, Anat Bremler-Barr and Hanoch Levy," *Vulnerability of Network Mechanisms to Sophisticated DDoS Attack"*, IEEE Transaction on computers, May 2013.

5. Zhang Fu, Marina Papatriantafilou, and Philippas Tsigas," *Mitigating Distributed Denial of Service Attack in Multiparty Application in the Presence of Clock Drifts",* IEEE Transaction on dependable and secure computing, May/June 2012.