ENHANCING THE IMPREGNABILITY OF LINUX SERVERS

Rama Koteswara Rao G[1], Satya Prasad R[2], Pathanjali Sastri A[3] and P. E. S. N. Prasad[4]
[1 & 3] V. R Siddhartha Engineering College, India
[2] Acharya Nagarjuna University, India
[4]Prasad V. Potluri Siddhartha Institute of Technology, India

ABSTRACT

Worldwide IT industry is experiencing a rapid shift towards Service Oriented Architecture (SOA). As a response to the current trend, all the IT firms are adopting business models such as cloud based services which rely on reliable and highly available server platforms. Linux servers are known to be highly secure. Network security thus becomes a major concern to all IT organizations offering cloud based services. The fundamental form of attack on network security is Denial of Service. This paper focuses on fortifying the Linux server defence mechanisms resulting in an increase in reliability and availability of services offered by the Linux server platforms. To meet this emerging scenario, most of the organizations are adopting business models such as cloud computing that are dependant on reliable server platforms. Linux servers are well ahead of other server platforms in terms of security. This brings network security to the forefront of major concerns to an organization. The most common form of attacks is a Denial of Service attack. This paper focuses on mechanisms to detect and immunize Linux servers from DoS .

KEYWORDS

Denial of Service, SYN Flooding, TCP Sequence Number attack, Brute Force attacks, Smurf attacks.

1. INTRODUCTION

Denial of Service attack is an attack that damages a server's hardware and software resources that is initiated by a person or any other system. These resources can be operating system data structures [2]. It makes a server unreachable and prevents end users accessing services of the server, modify system configuration information and can even destroy physical network components. These attacks disable a network, cause loss of data and results in financial losses to an organization. The risk of Denial of Service attack is unavoidable. DoS attacks are always malicious and illegal. Well-known popular web sites are repeatedly struck down by malicious hacker. To defeat detection the attacker can easily manipulate their traffic and the problem of identifying attack will be very difficult [6]. As per the survey conducted by FBI, these attacks are dreadful attacks in terms of financial losses for the organizations after information thefts [12]. As DoS attacks have become more regular, the DoS problem has inspired an mass of research into solutions[21]. The Denial of Service attack is one of the most common security threats and is also the most difficult problem faced in SOA. The SOA security must ensure that the legitimate users are not blocked from accessing the computing resources. The Denial of Service attack is one of the most difficult security problems faced and the SOA security must consider this to block the availability of a computing resource from the legitimate users of that resource.

This paper focuses on preventing DoS attacks from harming Linux servers. Common forms of Denial of Service (DoS) attacks include TCP SYN flooding attacks, TCP Sequence Number

Attack, Brute Force attacks and ICMP Smurf attacks. Routers, Web Servers, Web Services, Email Servers, etc. are most common targets for these form of attacks resulting in unavailability or degrade in performance of a particular service or the entire range of services offered by the targeted device.

2. Solution to identify and block the DoS attacks in Linux Kernel .

Current work focuses on the packet filtering rules that are defined in the firewall/router to identify and block the attacks. These rules monitor the network traffic, its source, its destination and its protocol type. This work focuses on

- Identifying attacking network and blocking it.
- To develop security measures on the Server.
- Analyzing number of times each IP connected to the Server.
- Monitoring Load on the CPU
- Check if the Server is flooded with SYN requests.
- Check the Server flooded with ICMP echo requests.
- Check if any DoS attack is targeting the Server.

3. TCP SYN flooding attacks

One of the most severe forms of attack is TCP SYN Flood attack because legitimacy of a client cannot be established during a TCP SYN Flood attack. Once the target host's resources are tired, no more incoming TCP connections can be recognized, thus denying further legitimate access [7]. During SYN flood attacks, the attacker sends SYN packets with non existing source IP addresses [14].

In SYN attack the client uses faked IP address to sends SYN messages to the Server. The server sends an ACK message that is for no reason returned. The server uses up its resources in the process while waiting for the ACK message from the client. The server becomes slow or insensitive to the other clients when the server is loaded with ACK messages. Flooding spoofed SYN requests can easily fail the victim server's backlog queue, causing all the arriving SYN requests to be dropped[10]. These attacks are unsafe, put away the server and make the websites and networks on the server unapproachable. In the critical real-time services the server may be slow or shutdown or kills valuable resources due to flooding of packets by SYN Flood attacks [15].

3.1 Detecting a SYN flood Attack

The client cannot receive a SYN/ACK packet from server by sending a packet from compromised client [13]. The SYN flood attack will not allow the Server to receive expected ACK code. The symptom of SYN Flood attack on the server is that the performance of the server will be slow. For Example the web site on the server will take long time to load or loads some elements of the page but not all. Attacker can perform SYN Flood attack by sending large number of SYN_RECV packets from a single IP address. This problem can be solved by adding the IP in firewall to stop the attack.

3.2 Defending from a SYN Flood Attack

The following steps are employed to defend from a SYN Flood Attack.

- Allow the server to avoid reducing connections when the SYN queue fills up.
- Increase the SYN backlog queue size.
- Reduce SYN_ACK retries.
- Reduce the SYN flood attack by lowering the timeout value for SYN_RECV.
- Protect IP Spoofing which is used for SYN Flood attack.

3.3 Algorithm for detecting and protecting from a SYN Flood Attack

```
For each packet arrival do
Check the IP Header Length
if IP Header Length = 20 then
 if protocol = TCP then
        If ipaddress is available in server side database then
                Packet is Correct.
              Forward the packet to destination.
        Else
            Check for symptoms of a SYN flood attack if not
              a known bad address.
            if SYN FLOOD ATTACK then
                Notify the administrator about SYN Flood
                  Attack.
                Identify the source of the packet and deny
                  the packet.
                If newsource then
                  Store the source of the packet
                    (IPaddress)in bad address database.
              End if
            End if
        End if
  End if
End if
End
```

4. TCP Sequence Number Attack

One end of the TCP session is controlled by the attacker. The attack will be successful when the attacked end of the network is tricked for the duration of the session. Attacker can respond to the sequence number similar to one used in the original session to disrupt the session The attacker obtains the system connection and gets the data form it by guessing the valid sequence numbers. The problem on TCP protocol is to initiate TCP sequence number spoofing through predicting TCP initial sequence number[9].

The communication session between the target and the trusted host can be exploited by the TCP sequence number.

4.1 Defending from a TCP Sequence Number Attack

The best idea is to allow the gateways to reject the external packets into the local net that claim to be from the local net. If the packet with the internal source addresses arrives on the external

interface deny it and to stop the attacks originating from the site, block the packets with a source address different from the internal network. One of the best practices to defend this attack is using TCP stacks with less predictable Initial Sequence Numbers (ISNs).

4.2 Algorithm for defending from TCP Sequence Number Attacks

```
For each packet arrived at the gateway do
        Check for the incoming interface.
        If source interface is external then
                Check for source address spoofing.
                If source address is internal then
                        Drop the packet.
                Else
                        Forward the packet.
                End if
        Else
      Source interface is internal, check for spoofed packet.
                If source address is external then
                        Drop the packet.
                Else
                        Forward the packet.
                End if
        End if
End
```

5. Brute Force Attacks

Security modules in Linux Servers use authentication to grant access to the services offered by the server. These modules are based on passive or active authentication processes. Passive methods require pre-determined inputs such as pre-shared keys and active methods require explicit user inputs such as passwords. A brute force attack employs software techniques to guess the correct combination of the inputs required by an authentication process.

The strength of any input required by an authentication process can be determined by the following criteria.

- The length of the input required.
- The expanse of the character set involved.
- The susceptibility of the input being related to meaningful information such as dictionary words, family names, dates, addresses etc.

The success rate of penetrating the built-in defence mechanisms in the Linux Servers depends on the availability of computing resources and the time required in performing a brute force attack.

5.1 Detection and Prevention

A security threat neutralizing mechanism involves detection and prevention. During the detection phase the threat can be identified by its type and/or source. In computer networks the threats can be localized or distributed resulting in a performance degrade or non availability of Linux Servers.

For instance a server running an FTP service may experience the following threat scenario.A single attacker repeatedly attempts to login to an FTP service without success.

5.2 An algorithm to detect this type of threat is presented below

- Use a predefined measure for the average number of running FTP processes
- Compare this with the number of running FTP processes.
- If there is a significant deviation, check the failure rate for the login action.
- When the failure rate is higher than expected we have a possible intrusion attempt.

5.3 An algorithm to prevent this type of threat is presented below

- Determine the source of the attacker such as an IP address or a MAC address.
- Block future attempts from the same source indefinitely or for a fixed period of time.
- Log the blocked sources and use ageing mechanisms to increase/decrease the duration of blocking period.

6. ICMP Smurf attacks

Attacker uses fake IP address to send messages to a computer as if the messages are coming from a trusted host. The main aim of smurfing is to conceal sender's identity. Smurfing makes use of Internet Protocol (IP) and Internet Control Message Protocol(ICMP). Smurf attack floods a system via spoofed broadcast ping messages. The attacker sends large number of ICMP echo requests with spoofed source IP addresses to IP broadcast addresses. The hosts on this network will accept the ICMP echo request and reply to it with an echo reply. Due to large amount of echo replies from multiple hosts would consume large amount of network bandwidth and result in slow down of network. Any broadcast enabled network or any host responding to broadcast address can be a potential target for ICMP smurf attacks. At the intermediary network by disabling the IP-directed broadcast service can block the smurf attack[11].

6.1 Defending from ICMP Smurf Attacks

Following are the steps to prevent ICMP Smurf attacks.

- The hosts and routers need to be configured not to respond to ping requests or broadcasts.
- Routers need to be configured not to forward packets directed to broadcast addresses.
- Do not allow your firewall to accept ICMP echo requests from the Internet.
- Restrict the flow of information outbound from one network to another to ensure that Smurf attack is not launched.
- Simply block all inbound and outbound ICMP *echo-request* and ICMP *echo-reply* packets.

6.2 Algorithm for defending from ICMP Smurf Attacks

For each packet arrived at the router do
Check for ICMP echo messages
If type of packet is ICMP echo-request or ICMP
echo-reply then
If configuration-rule is to allow then

```
                    Check for destination address.
                    If ipdestination is subnet or
                       broadcast then
                        Drop the packet.
                         Log the event.
                    Else
                          Forward the packet.
                    End if
               Else
                    Drop the packet.
               End if
          Else
               Forward the packet.
          End if
End
```

7. Statistical Comparison between different types of DoS attacks

DoS attacks have no boundaries and have hit all the sectors of the industry such as financial services, banking, insurance, hospitality, travel, government organizations, defence, etc. The attacks mainly focus on bandwidth capacity and routing infrastructure. It has been observed that most used DoS attack is TCP SYN Flood as summarized in the table below.

Type of Attack	**How attack works**	**Impact**	**DoS Attacks Most Used (Approximately)**
ICMP Smurf Attack	Floods a system via spoofed broadcast ping messages	Causes the victim's server to crash.	27%
TCP Hijacking	Attack may be used to gain illegal access to system resources.	All Unencrypted TCP protocols are susceptible for this type of attacks.	22%
TCP Sequence Number Attack	Attacks exploit the communication session, which was established between the target and the trusted host that initiated	This attack disrupts or hijacks a valid session.	21%

	the session.		
TCP Syn Flood Attack	Exploits weakness in TCP/IP protocol.	Affects the resources that run TCP Server Processes.	30%

Table 1 : Statistical Comparison between different types of DoS attacks

8. Disabling Highly Vulnerable Services

To improve the overall performance of the system, we need to enhance the security by prioritizing the services on a risk based assessment and disabling the services with highest threat vulnerability .

Port Nos	**Port Names**	**Risk**
20, 37, 80, 110, 119, 161, 443, 465	FTP-DATA, Time service, HTTP, POP3, NNTP, SNMP, HTTPS, SMTP	Low
22,143	SSH,IMAP	Low to Moderate
23,25	Telnet, SMTP	Moderate
21,53,512	FTP server port, DNS, TCP	High

Table 2: Classification of ports based on risk vulnerability

Service	**Remarks**
NFS	NFS and its related services like nfsd, mountd, portmapper, lockd, etc are dangerous over the Internet.
Remote Shell Services	The remote commands like rsh, rexcec and rlogin are most dangerous if these commands are exposed to the Internet.
Telnet server	Use sshd instead of Telnet
FTP	FTP can be replaced with http. Files can be exchange in better way using http.

Table 3: Classification of services based on risk vulnerability

9. Safety measures to prevent different types of DoS attacks

Key kernel parameters can be adjusted to immunize Linux systems (for instance, in the file /etc/sysctl.conf) from typical DoS attacks to a certain degree as summarized in the following table.

Attack	Parameters	Recommended value
TCP SYN Flood Attack	Using SYN cookies	Enable
	Increasing the SYN backlog queue	2048
	Reducing SYN_ACK retries	3
	Setting SYN_RECV timeout	40
	Preventing IP spoofing	Enable
TCP Sequence Number Attack	Forwarding of source routed packets	Disable
TCP hijacking Attack	Source ports	32768 to 61000
ICMP Smurf Attacks	Ignore Smurf attacks	Enable
	Forward traffic between interfaces	Disable
	Protect against SYN flood attacks	Enable
	ICMP Redirect Acceptance	Disable
	Burst-normal value	1 to 100000
	Burst-max value	1 to 100000
	Lockup-value	1 to 10000

Table 4: Safety measures to prevent different types of DoS attacks

10. Results

Network traffic observed at the server before & after implementation of defence algorithm during a simulated SYN Flood attack

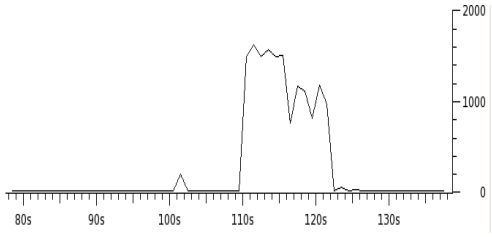

Figure 1

Figure 1 indicates that the server was attacked by the SYN Flood attack at approximate Time (T) 110secs. It has been observed that the number of packets captured at the network interface of the server machine is more than 1500 packets per second during the SYN Flood attack. This indicates that the SYN Flood attack consumes network bandwidth and resources on the server machine. When the implementation of algorithm 3.3 is executed at Time (T) 122secs the number of packets captured by the machine have been dropped to less than 10 packets per second which is in the range of current system normal load.

Network traffic observed at the server before implementation of defence algorithm during a simulated ICMP Smurf attack

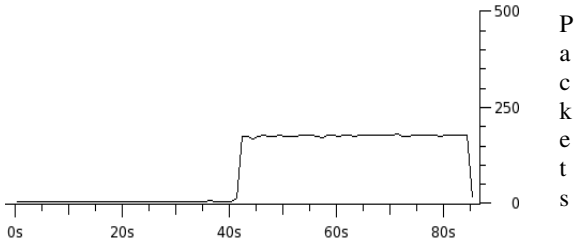

Figure 2

Figure 2 depicts network traffic analysis at the server machine before and during a simulated ICMP Smurf attack. It has been observed that while the average number of packets before the attack occurring at approximate Time (T) 41secs was around 10 packets per second, the number of packets captured at the network interface of the server machine is more than 200 packets per second during the ICMP smurf attack.

Network traffic observed at the server after implementation of defence algorithm during a simulated ICMP Smurf attack

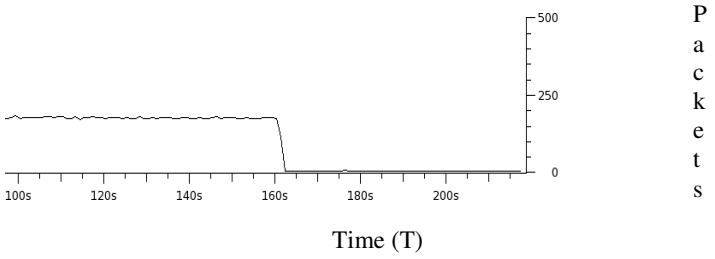

Figure 3

Figure 3 depicts network traffic analysis at the server under the simulated ICMP Smurf attack after execution of the implementation of algorithm 6.2. The total number of packets received per second on the server after defending the attack at approximate Time (T) 161secs on the server by applying algorithm 6.2 (i.e by dropping the packets if the type is ICMP echo request or ICMP echo reply) has stabilized the network traffic back to the earlier observed normal operating load.

11. CONCLUSIONS

Global businesses are shifting towards emerging technologies such as service oriented architecture using cloud computing for a sustained revenue model. Ensuring round the clock uninterrupted service to the clients thus, becomes a top priority to any organization. Denials of Service attacks cause significant losses in terms of time and money for the affected organizations. This paper discuses the different types of Denial of Service attacks specific to Linux server platforms and presents ways to detect and prevent such attacks or to avoid such attacks altogether so that the server platforms that form the backbone of service oriented architecture function smoothly without a breakdown.

References

[1] A. C. Snoeren, "Hash-based IP Traceback," In Proceedings of the ACM SIGCOMM Conference, 2001, pp. 3-14

[2] Bahaa Qasim M. AL-Musawi, " Mitigating DoS/DDoS Attacks Using Iptables", International Journal of Engineering & Technology IJET-IJENS Vol: 12 No: 03,June 2012.

[3] B. B. Gupta, R. C. Joshi, and Manoj Misra, "Distributed Denial of Service Prevention Techniques, International Journal of Computer and Electrical Engineering", Vol. 2, No. 2, April, 2010.

[4] Bin Xiao,Wei Chen,Yanxiang He, "An autonomous defense against SYN flooding attacks: Detect and throttle attacks at the victim side independently", Journal of Parallel and Distributed Computing, Elsevier, pp. 456-470,July 2007.

[5] Charalampos Patrikakis, Michalis Masikos, and Olga Zouraraki, "Distributed Denial of Service Attacks", The Internet Protocol Journal, No. 4, ISSN: 1944-1134, December 2007.

[6] Chen-Mou Cheng, H.T. Kung, Koan-Sin Tan, "Use of Spectral Analysis in Defense Against DoS Attacks", Proceedings of IEEE GLOBECOM 2002,pp. 2143-2148 vol.3,November 2002.

[7] Christoph L. Schuba, Ivan V. Krsul, Markus G. Kuhn, Eugene H. Spafford, Aurobindo Sundaram, Diego Zamboni, "Analysis of a Denial of Service Attack on TCP", Proceedings of the 1997 IEEE Symposium on Security and Privacy, pp.208.

[8] C.L. Schuba, I.V. Krsul, M.G. Kuhn, E.H. Spafford, A. Sundaram, D. Zamboni, Analysis of a denial of service attack on TCP, in: Proceedings of the IEEE Symposium on Security and Privacy, IEEE Computer Society Press, Silver Spring, MD, 1997, pp. 208–223.

[9] Fanping Zeng , "Research on TCP Initial Sequence Number Prediction Method Based on Adding-weight Chaotic Time Series",Proceedings of IEEE, ICYCS 2008, pp. 1511-1515.

[10] Haining Wang ,Danlu Zhang and Kang G. Shin, "Detecting SYN Flooding Attacks", Proceedings of IEEE INFOCOM 2002, pp. 1530-1539, June 2002.

[11] Jaydip Sen, "A Robust Mechanism for Defending Distributed Denial of Service Attacks on Web Servers", International Journal of Network Security & Its Applications, Vol.3, No.2, March 2011.

[12] Kumar K., Joshi R., and Singh K., "An Integrated Approach for Defending Against Distributed Denial of Service Attacks," http://www.cs. iitm.ernet.in/~iriss06/paper.html, 2002.

[13] L. Yun, Ye, G., & Guiyi, W., "Detect SYN Flooding Attack in Edge Routers," International Journal of Security and Its Applications (IJSIA), vol. 3, pp. 31-45, 2009.

[14] L. Kavisankar and C. Chellapan, "Challenging Number Approach for Uncovering TCP SYN Flooding using SYN Flooding Attack", International Journal of Network Security & Its Applications (IJNSA), 3, No. 5, Sep 2011.

[15] Mehdi Ebady Manna and Angela Amphawan, " Review of SYN-FLOODING Attack Detection Mechanism", International Journal of Distributed and Parallel Systems (IJDPS) Vol.3, No.1, January 2012.

[16] Mitko Bogdanoski and Aleksandar Risteski, “Wireless Network Behavior under ICMP Ping Flood DoS Attack and Mitigation Techniques”, International Journal of Communication Networks and Information Security, Vol. 3, No. 1, April 2011.

[17] [R4] M.Voznak and J. Safarik, ”DoS Attacks Targeting SIP Server and Improvements of Robustness”, International Journal of Mathematics and Computers in Simulation, issue 1, volume 6, 2012.

[18] Noureldien A. Noureldien, Mashair O. Hussein, “Block Spoofed Packets at Source (BSPS): A method for Detecting and Preventing All Types of Spoofed Source IP Packets and SYN Flooding Packets at Source: A Theoretical Framework”, International Journal of Networks and Communications 2012, pp. 33-37.

[19] Oliver Zheng, Jason Poon, Konstantin Beznosov, “Application-Based TCP Hijacking”, EUROSEC '09 Proceedings of the Second European Workshop on System Security, pp. 9-15.

[20] Pukkawanna, V.Visoottiviseth, P.Pongpaibool, "Lightweight Detection of DoS Attack", In Proc. of IEEE ICON2007, Adelaide, South Australia, November,2007.

[21] Tom Anderson, Timothy Roscoe, and David Wetherall, “Preventing Internet Denial-of-Service with Capabilities”, Intel Research Berkeley, Intel Corporation, Copyright 2003.