International Journal of Advanced Trends in Computer Applications
www.ijatca.com

An Enhanced Model for Mitigating DDoS Attacks on Linux Servers using IPTables and Bash scripts

[1]Nwachukwu V. C, [2]Ikerionwu C. O, [3]John-Otumu A. M
[1, 2, 3]Department of Information Technology, Federal University of Technology, Owerri, Nigeria
[1]victor.nwachukwu@futo.edu.ng, [2]charles.ikerionwu@futo.edu.ng, [3]adetokunbo.johnotumu@futo.ng

Abstract: *A distributed denial-of-service (DDoS) attack is one of the most powerful weapons on the internet. Research indicates that several works been have done to mitigate DDoS attacks on Linux based Servers. However, the type of DDoS attack covered were mostly HTTP Get Flood attacks on port 80 and 443. More so, the IPTables firewall rules used were not automated using Bash scripts to make it portable and the firewall rules in most cases were written to mitigate attacks coming from a single IP address. This study will therefore expand the scope of the mitigating DDoS attacks using IPTables to include TCP SYN Flood attacks, UDP Flood attacks and PING (ICMP) Flood attacks. After carrying out the test when the BASH scripts have been executed, DDoS attacks in form of TCP SYN Flood, UDP Flood and ICMP (Ping) Flood were generated using HPing3 and they were successfully mitigated as the Linux Server dropped packets that make up these attacks while allowing legitimate traffic and users to access resources on the Server.*

I. INTRODUCTION

On proactive defense during wars, Thomas (1988) opined that, one should not rely on the likelihood of the enemy not coming but on one's own readiness to receive the enemy, not on the chance of the enemy not attacking but rather on the fact that one's position is unassailable. In the world of Computing and Information Technology, there are two typesof people namely: people who care about security and people who should care about security. In other words, information is an asset that has a value like any other asset. According to Mihalos, Nalmpantis&Ovaliadis (2019), Information security has become indispensable because of the distinctive value that data has gained in our days. Securing data and information from attacks has become imperative. This means to maintain the Confidentiality, Integrity and Availability (CIA) of data and information.

Most Cyber-attacks are launched to breach the security framework and hijack valuable data and information. Denial of Service (DoS) attacks attempt to deny legitimate users access to services. The Distributed Denial of Service (DDoS) aims at consuming important resources available on a Server or a Router so that the capacity of these systems to provide legitimate services are diminished or completely exhausted. The attacker floods the victim's machine with excess traffic large enough to exhaust the disk, saturate the connection link or overflow the communication buffer (Amadi et al., 2015). In most DDoS attack tactics, numerous attack agents are utilized to target system almost at the same time with the aim of completely consuming the resources of the target system (Cho, Kim, Lee, & Lee, 2015). Similarly, malformed packets can be sent to disrupt an application, service or a protocol available on a victim's machine in order to force a reboot or completely make the machine unresponsive (Mirkovic and Reiher, 2004).

DDoS attacks in recent years have been targeted at corporate bodies and government agencies some of which are attributed terror groups and rival nation's intelligence agents (Amadi et al., 2016). Examples of such will include the numerous cyberattacks on the White House and other United States of America (US) assets and agencies, the China-Google faceoff, Wiki leaks, conflict between Russia and Estonia.

Due to the conspicuous nature of a successful DDoS attack, it is a common cyber-attack choice for cyber warfare (Waziri, 2016). Given the likelihood of a DDoS attack, a defense mechanism adopted should drop malicious packets sent by an attacker with high accuracy, while utilizing minimal system resource as well as low false positive scores (Simon, Huraj&Cernansky, 2015).

Identifying malicious nodes and denying packet injections from such sources is the key objective of a DDoS defense approach (Wu et al., 2010). The firewall

is the common defense mechanism used in network security (Mihalos et al., 2019). Firewalls prevent intrusions and provide high level of defense against illegitimate activities (Rehman et al., 2010). Firewalls are network security tools that operate between the connection of an organization's internal and the external network. Firewall's philosophy is basically to build a barrier at this choke point where all incoming and outgoing traffic passes.

The philosophy of a firewall is to create a barrier where incoming packets and outgoing packets will pass through as illustrated in Figure 1. Rules are defined in this barrier which decides the fate of each packet as they pass through the firewall. Packets that do not match the stated rules are blocked and prevented from entering the site's network.

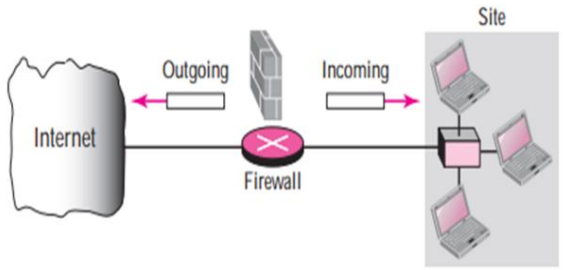

Figure 1: Firewall Mechanism (Behrouz, 2010)

Firewalls built on Linux Operating System (OS) architecture are robust, inexpensive, customizable to a high extent and versatile (Lucian, 2006). Linux Firewall solutions can be free such as IPTables available on Linux platforms or can come as preconfigures hardware firewalls solutions such as Cisco and Juniper firewalls (Konikiewicz et al., 2017). To build firewalls using Linux OS, Netfilter and IProute packages are needed. Netfilter is the framework for packet filtering, IProute provides advanced routing, both of which are managed using IPTables (Wu, 2012).

The literature indicates that several works have been done to mitigate Linux Server attacks, for example, Mustafa and Suraiya (2016); Bhisham and Karan (2011); Bahaa (2012); Mihalos et al. (2019); and Simon et al, (2015) are among the notable contributions. However, findings show that the approach adopted in these studies have some important limitations. While the type of protocol filtered are mostly Hypertext Transfer Protocol (HTTP), existing research have focused on mostly HTTP Get flood attack. Furthermore, the implementation and usage of the proposed firewall rules require high-level expertise, such that only IPTables savvy server administrators can execute; basic and routine configurations are not automated. The existing research also lacks the functionality of keeping track of all events handled by the server administrator using the firewall script.

II. RELATED WORKS

The Linux-based firewall, IPTables is part of the Linux Operating System (OS) since 2001 and it has grown into a powerful firewall posing most functionalities obtainable in commercial firewalls (Rash, 2007). Some of those functionalities include: rate limit, filtering policy, stateful packet tracking among others. These days, IPTables is a mainstay in major Linux OS variants.

Adwitiya, Srinidhi & Vignesh (2016) compared IPTables firewall solutions with Windows and Hardware firewalls shown in Table 1. The comparison highlights the advantage of IPTables in attributes such as customization, automation and cost efficiency.

Wenhui & Junjie (2013), after conducting series of experiments with Cisco ASA 5505 and Linux IPTables concluded that, the difference between the two firewall products is quite obvious as illustrated in Figure 2. The authors noted that 8,000 requests, the performances of ASA 5505 and IPTables are the same. Between 8,000 and 10,000 requests, there is seldom failure occurrence for both Cisco ASA and IPTables. Beyond 10,000 requests, the rate of failure gradually increase on ASA 5505 while IPTables remains unchanged. Failure rate of Cisco ASA got over 160 corresponding to 15,000 requesting clients while Linux IPTables firewall remained less than 10.

Table 1: Comparison of Firewalls and IP-tables (Adwitiya, Srinidhi & Vignesh, 2016)

Properties	Windows firewalls and Hardware Firewall	IPTables Firewall
Cost efficiency	Available for pricing or high cost	Free with all flavors of Linux OS
Configuring	Works with IP and port blocking	Works with each application
Efficiency	Compromises are possible	Highly efficient
Customization of Firewalls	Partial endorsement	Fully customizable
System administrator intervention	Needs Administrator supervision to take care so that nothing goes wrong	Can be automated by postscripts and custom rules

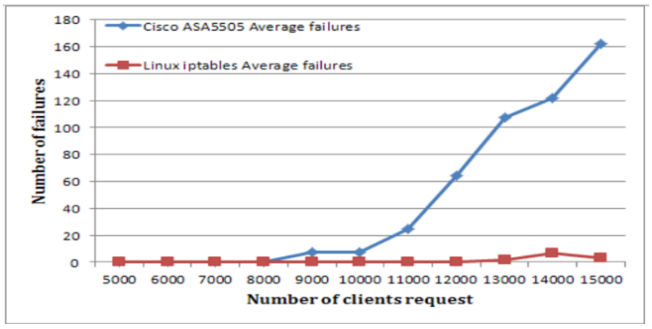

Figure 2: Comparison of Cisco ASA firewall and Linux IPTables Firewall (Wenhui & Junjie, 2013)

IPTables is made up of a table of chains. These chains define how the kernel handles packets. In addition to

tables and chains, IPTables have matches and targets (Russel, 2002). Chains are created by rules which are to be applied on packets as they pass through the firewall. Input, Output, Forward, Pre-routing and Post-routing made up the predefined chains in IPTables (Linde, Pumputis, &Rodr, 2015)

IPTables targets are ACCEPT, DROP or REJECT. Each rule contains one of these targets which determines the fate of packets as the pass the firewall. ACCEPT target accepts packets, DROP target drops packets without sending a notification to the sender while REJECT target drops packets but notifies the sender (Sara, 2018). Packet filtering in IPTables is through packet header fields and actions to be taken are determined by the targets (Al-musawi, 2012). IPTables complexity largely depends on the purpose for which the rules are being written (Chatterjee, 2013).

Amongst various online attacks hampering IT security, DDoS attacks ranks among the most devastating to firms and government agencies. Security experts now face overwhelming pressure in recent times, thus creating a need for the development of more effective defense solutions. (Emmanuel, 2018).

Rehman & Rahman (2010) proposed ZoneAlarm as the best security result against all attacks. The authors concluded that, in order for a firewall to be properly configured, the user interface should be simple and attractive. ZoneAlarm and Comodo firewall are easy to configure, given its user-friendly interface. ZoneAlarm however costs as much as £54.95 for a Year and can allow up to five devices (ZoneAlarm, 2021). It is also built only for Microsoft Windows Operating System. Zone alarm is application-level firewall (Baha, 2015), making it unsuitable for Layer 3 and 4 attacks.

Emmanuel (2018), in his work used a game theory approach to determine the optimal setting for a firewall to mitigate against rogue packets. His work however, focused on DDoS attacks on network devices with emphasis on web servers and the provision of security to the web server was limited to ports 80 and 443.

Waziri (2016) designed a framework implementing a virtual and a hardware firewall, the two firewalls were connected in parallel. A monitor was built to keep track of the firewall state and to execute certain commands according to the states of the firewall. This monitor basically redirects packets when there is an active HTTP flood attack which have overwhelmed a firewall. His work however, focused on HTTP GET Request Flood and the hybrid topology will have a negative impact on network performance when thousands of packets will have to transverse two firewalls to make it to the destination.

Šimon, Huraj&Čerňanský (2015) experimented a testbed environment based on a peer-to-peer (P2P) grid for DDoS attacks. IPTables tool was used for packet filtering and consequently for preventing DoS/DDoS attacks. In their experiment, IPTables effectively mitigated the DDoS attack (HTTP Get flood attack), decreasing the volume of data sent and received. Their work however, focused on HTTP Get flood attack and the testbed environment was designed using OS Debian7 which has reached its End-of-life (EOL)

Qasim & Al-Musawi (2012), developed IPTables firewall rules to mitigate DoS attacks. However, their firewall script identifies legitimate or illegitimate traffic based on source address, destination address and protocol type. This limits the firewall to only DoS attacks as the rules basically block packets coming from a suspected IP address.

Chatterjee (2013) implemented a firewall using IPTables inside a virtual Local Area Network (VLAN). The firewall rules were used to sieve packets to minimize DDoS attacks. The IPTables firewall rules were however limited to TCP SYN flood attacks coming from a particular IP address.

Deshpande (2015) designed a network environment where a Honeypot was installed to capture malicious traffic while allowing legitimate traffic get into the internal network. Despite all the advantages, Honeypot has a few shortcomings as the organization's router was still being flooded by unwanted traffic which will reduce the overall performance of the router. This will have a ripple effect on the organization's internal network.

Mustafa and Parveen (2016), used IPTables as firewall to mitigate DoS/DDoS attacks generated during their experiment. They presented IPTables as a simple and economical tool for combating DDoS attacks. However, their work focused on SYN flood attacks and did not consider spoofed addresses.

Ramkumar and Subbulakshmi (2021) , used IPTables to mitigate TCP SYN Flood. However, they did not consider spoofed addresses as well as ICMP and UDP flood.

III.METHODOLOGY

Setting up the virtual environment in Figure 3, involved the following steps:

1. Windows OS was installed on a Workstation. The Workstation's specification is 12GB (Random Access Memory) RAM, 1 Terra Byte (TB) Hard Disk (HDD), Intel core i7 processor with NVIDIA Geforce graphic adapter.

2. VMware was installed on the workstation. VMware enables users to installed virtualize Operating Systems while running on a host machine. VMware also creates virtual network adapters which users can use to monitor network traffic as it is obtainable in real life scenarios.

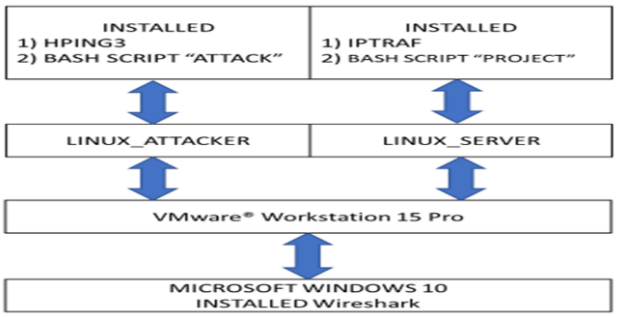

Figure 3: Virtual Environment

3. Two CentOS Linux Severs were installed on the VMware. One will serve as the Linux Server while the other will be Linux Attacker as illustrated in the Figure 3.1.

4. Wireshark Network Analyzer software and ColasoftCapsa Network monitoring software were also installed on the Windows Workstation to monitor packets coming from the Linux Server and the Linux attacker. The monitor monitoring tools installed used the virtual network adapter created by VMware, VMnet, to capture network traffic created by the two virtualized Linux machines

5. HPing3 was installed on Linux Attacker. This tool was used to generate packets at a volume enough to simulate a DDoS attack. These packets were directed to the Linux Server.

6. IPTraf was installed on Linux Server to monitor the incoming and outgoing rate for packets passing through its interface.

7. Bash scripts "Attack" automated the process of generating rogue packets using HPing3. This script was developed and installed in Linux Attacker while Bash script "Project" was developed and installed in Linux Server. "Project" was designed to mitigate those rogue packets forming DDoS attacks coming from the Linux attacker

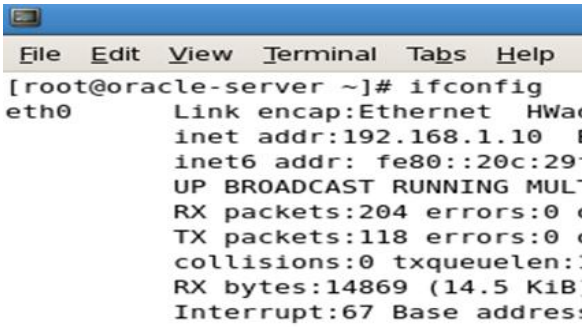

Figure 4: Network Status of Linux Sever

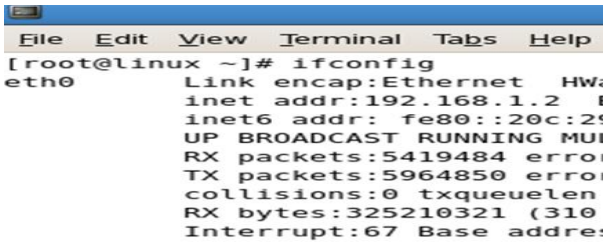

Figure 5: Network Status of Linux Attacker

Figure 4 shows network status of the Linux Server with IP address 192.168.1.10. Figure 5 shows network status of the Linux Attacker with IP address 192.168.1.2. The addresses were made to have the same network path so the two virtualized Linux Machines can communicate with each other. This can be verified using the PING command as shown in Figure 5. In a Linux environment the network status will be displayed when the "ifconfig" command is entered on the CLI (Command Line Interface).

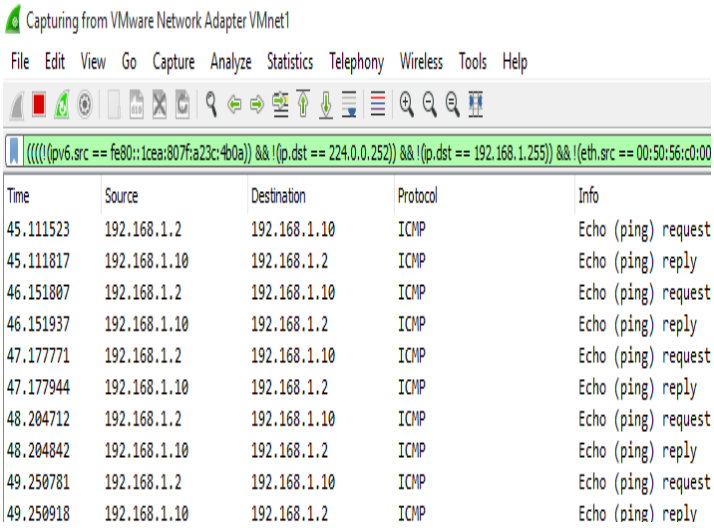

Figure 5: Wireshark capturing ping requests and replies from the Linux machines

To test the virtual network environment, ping requests were set from Linux Attacker to Linux Server, and Linux Server replied accordingly. These ping packets were captured by Wireshark as shown Figure 3.5. Also, to test if Linux Server accepts remote connections, a telnet request was made from Linux Attacker using "Telnet 192.168.1.10"as shown in Figure 6.

Figure 6: Telnet request from Linux Attacker

The telnet connection was successful, the TCP standard three-way handshake (SYN, SYN-ACK, ACK) was first initialized before the actual data transfer and the packets were captured using Wireshark network analyzer as shown in Figure 7.

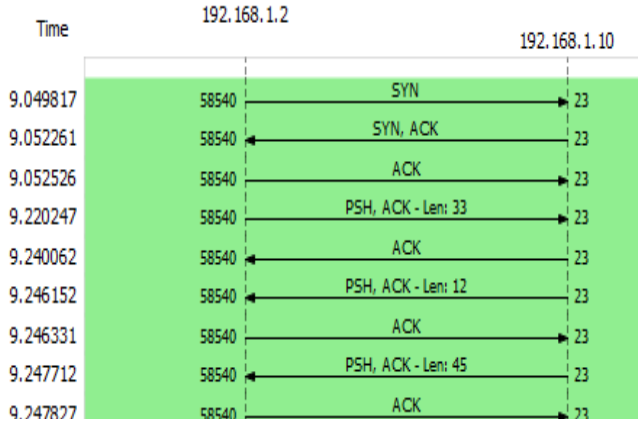

Figure 7: Wireshark capturing packets from telnet session

With these preliminary checks, the Linux machines have been properly installed and can communicate with each other. Thus, the virtual network environment is now ready so simulation process can now commence.
HPing3 was used to initiate the DDoS attack, generating and directing thousands of packets to Linux Server in order to simulate a live DDoS attack pattern as shown in Figure 8.

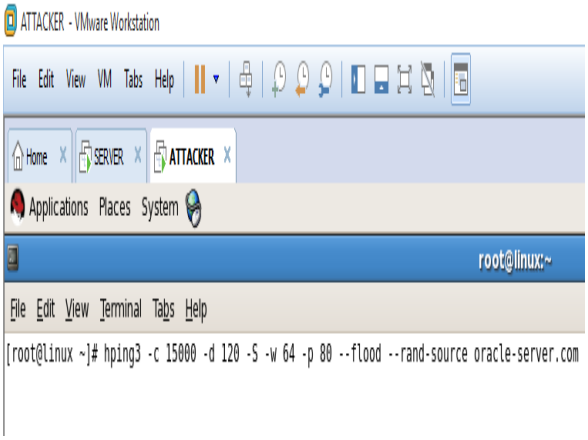

Figure 8: Initiating TCP SYN flood attack

HPing3 will generate thousands of TCP SYN packets and flood the Server through port 80 while randomizing the source addresses to mimic a DDoS attack which will involve the use of spoofed addresses and botnets. The network monitoring tools already installed were used to capture the packets with the DDoS attack still in session.

IV. RESULTS AND DISCUSSION

Three network monitoring tools Wireshark, Colasoftcapsa and IPTraf were deployed to monitor packets when TCP SYN DDoS attack was being simulated. Results from them will be used for the analysis.
The Wireshark flow graph as shown in Figure 9, illustrates packets flow during an active DDoS attack initiated by the Linux Attacker (192.168.1.2) and the Linux Server (192.168.1.10). The Linux Attacker sent the SYN packets and Linux Server replied with RST, ACK packets which indicates that the connection between the two have closed and subsequently, Linux Server now drops all SYN requests form Linux Attacker, freeing up enough bandwidth to serve legitimate users.

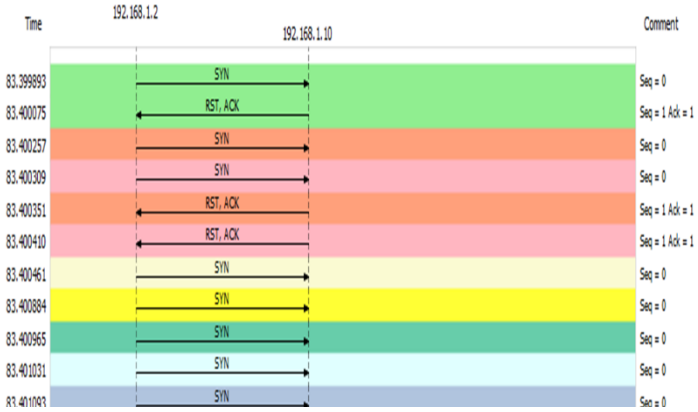

Figure 9: Wireshark flowgraph during a DDoS attack

ColasoftCaspa Network tool also captured the RST packet per second (pps) rates before and after the activation of the Bash script a shown in the Figure 10 and Figure 11 respectively.
It was observed that the average RST pps before was 21,250 pps and it reduced to one pps after the implementation of the Bash script. The other results from ColasoftCapsa network monitoring tool are displayed in table 2. These results have been grouped into different parameters and analysed as thus:

1. **Average Packets Per Second (pps)**: This dropped from 60,232 packets per second to 20,274 packets per second. $\frac{60232-20272}{60232} \times \frac{100}{1} = 66\%$. This indicates a 66% drop in pps. This result shows that the DDoS attack have been mitigated by two-thirds, freeing up enough bandwidth space for legitimate users to access the Linux Server.

2. **Total TCP ACK sent:** Another parameter that can be used to show the performance of DDoS mitigation script is the TCP ACK. Form Colasoftcaspa network tool, TCP ACK packets sent before implementing the script was 4,780,091 packets that is in response to 6,122,019 SYN packets.
Comparing Total SYN packets sent to Total ACK packets in ratio form will be

$6{,}122{,}019\ SYN\ packets : 4{,}780{,}091\ ACK\ packets$ which can be reduced to $1.28\ to\ 1$.

Thus, the ratio of SYN packets to ACK packets during an active DDoS attack was 1.28 to 1. This means that Linux Server was trying to acknowledge as many SYN request as possible even when they constitute a TCP SYN flood attack. At this rate, Linux Server will be unable to response to legitimate users, which is the goal of the attacker. However, when the DDoS mitigation script was implemented, TCP ACK packets fell from 4,780,091 packets to 194 packets. Percentage decrease can be derived thus

$$\frac{4{,}780{,}091 - 375}{4{,}780{,}091} \times \frac{100}{1} = 99\%$$

That is a massive 99% decrease in number of rogue SYN request being replied by Linux Server.

To compare the Total SYN packets sent and the total ACK packets, we now have

The ratio now stands at 16,327 to 1, i.e., for every 16,327 rogue TCP SYN packets sent, one will be replied with an ACK. The overall number performance will improve as the resources that were used to reply those rogue SYN requests are now being deployed to legitimate requests.

3. **Total TCP RST sent:** RST packets are used by the host to show that it will not accept or receive new connections. Before the implementation of the DDoS mitigation, total TCP RST sent by Linux Server was 4,517,907 in response to 6,122,019 SYN packets received from the DDoS attack. The ratio of SYN packets to TCP packets was 1.36 to 1. This ratio shows that the server can not respond new connections as the resources have been consumed by the attack. On implementing the DDoS mitigation script developed, the number of RST packet sent fell from 4,517,907 packets to 282 packets, which represents a 99% drop in the number of RST sent. This indicates that Linux server will accept new connections and thus legitimate users can access the server.

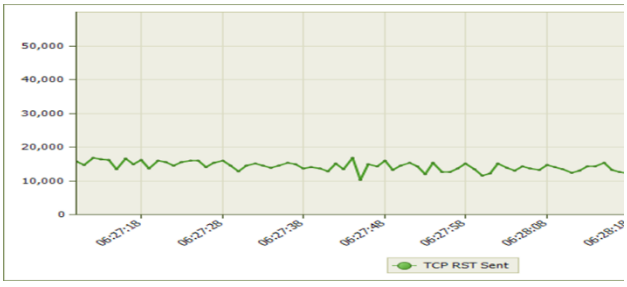

Figure 10: Colasoftcapsa TCP RST packet rate before implementing scripts

Figure 11: ColasoftCapsa TCP RST packet rate after implementing scripts.

Table 2: Summary from ColasoftCapsa network monitoring tool during TCP SYN FLOOD

Parameter	Before using the firewall script	After implementing the firewall script	Results/Comments
Average packets per second (pps)	60,232	20,274	66.3% decrease
Average bits per sec (bps)	45Mbps	29Mbps	35% decrease
Total TCP SYN sent	6,122,019	6,122,000	-
Total TCP ACK sent	4,780,091	375	99% decrease
Total TCP RST sent	4,517,907	282	99% decrease
Ratio of TCP SYN to TCP ACK	1.28: 1	16,327: 1	Less ACK packets were sent by the server during SYN flood attack
Ratio of TCP SYN to TCP RST	1.36: 1	21,695: 1	RST packets were dropped by the server
TCP SYN ACK Sent per second in packets per second (pps)	21,250 pps	1 pps	Improved network performance
TCP RST Sent per second in packets per second (pps)	21,052 pps	1 pps	Improved network performance

V. CONCLUSION

The following is the summary of the methods and procedures adopted to develop the mitigation technique in this work:

1. A BASH script was written and deployed on the Linux server. This script simplifies the process of monitoring system resources and settings by providing a UI which can be used to generate reports on the system.

2. Two algorithms were developed for the BASH script. Algorithm for the proposed system (see appendix A) was used to write the BASH script while Algorithm for DDOS mitigation was used to setup the IPTables rules. The BASH script and the IPTables rules make up the Anti-DDOS firewall.

3. The mitigation approach was simulated using VMware workstation (see figure 3.3) for illustration. The mitigation approach proved to mitigate DDoS attack effectively when Hping3 attack tools was used from centos Linux

The mitigation approach presented in this work integrated features like packet sending limits, IP source verification, packet drops mechanism that drops packets from a source judged to be an attacker at a particular time.

FUTURE WORK

Linux operating system is the backbone for this research work. Other operating systems deployed for servers can also be incorporated using the Anti-DDOS firewall algorithm developed in this work. Mobile operating systems are also included in this list.
Machine learning algorithms and Artificial Neural Networks (ANN) can also be explored to detect DDOS attacks so the Anti-DDOS firewall will only be activated when there is an active DDOS attack. This is will improve overall network performance.

REFERENCES

[1] Adwitiya M., Srinidhi S., Vignesh C. J. (2016), "An Analytical Study on the Versatility of A Linux Based Firewall From a Security Perspective", Department of Computer Science, Amrita Vishwa Vidyapeetham, Mysore Campus, Mysore-570026, Karnataka, India. Pg. 3-4

[2] Al-musawi, B. Q. M. (2012). Mitigating DoS / DDoS Attacks Using IPTables. June, 101–111. International Journal of Engineering & Technology IJET-IJENS Vol: 12 No: 03

[3] Amadi E. C, Ani E. E., Eke M. C., Jibiri J. E. (2015). An in-depth analysis of the possible approaches to detection and offensive defense of DDoS attack on network server, *3*(11). IJRIT International Journal of Research In Information Technology, Volume 3, Issue 11, November 2015, Pg. 36-51

[4] Amadi, E. C., Ajanwachuku, N. C., Nwachukwu, V., Anyalewechi, I., & Shandilya, D. (2016). A Review on the Application of Game Theory to Computer. 03(13), 842–849. International Journal of Research

[5] Behrouz A. F. (2010), TCP/IP protocol suite, The McGraw-Hill Companies, Inc. Fourth Edition

[6] Chatterjee, K. (2013). Design and Development of a Framework to Mitigate DoS/DDoS Attacks Using IPtables Firewall. International Journal of Computer Science and Telecommunication, 4(3), 67–72. http://www.ijcst.org/Volume4/Issue3/p11_4_3.pdf

[7] Cho, J., Kim, J., Lee, G., & Lee, H. (2015). DDoS Prevention System Using Multi-Filtering Method, 769–773. International Conference on Chemical, Material and Food Engineering (CMFE-2015)

[8] Deshpande, H. A. (2015). HoneyMesh : Preventing Distributed Denial of Service Attacks using Virtualized Honeypots. 4(08), 263–267. International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 Vol. 4 Issue 08, August-2015

[9] Emmanuel, A. (2018). A Game Theory Model for Detection and Mitigation of DDoS Attacks on Web Servers. Department of Information Management Technology, Federal University of Technology, Owerri

[10] F., Konikiewicz, I. W., Markowski, M., Wyspianskiego, W., Networks, C., &Wyspianskiego, W. (2017). Analysis Of Performance and Efficiency OfHardware And Software Firewalls, *9*(1), JACSM 2017, Vol. 9, No. 1, pp. 49 - 63 10.1515/jacsm-2017-0003. Department of Systems and Computer Networks, Wroclaw University of Science and Technology, Wroclaw, Poland

[11] Linde, P., Pumputis, M., & Rodr, G. (2015). iptables revisited : a not so ordinary firewall.

[12] Lucian G. (2006), "Designing and Implementing Linux Firewalls and QoS using netflter, iproute2, NAT, and L7-flter", 1st Edition, Packt Publishing.

[13] M. G. Mihalos, S. I. Nalmpantis, K. Ovaliadis (2019), "Design and Implementation of Firewall Security Policies using Linux Iptables", Journal of Engineering Science and Technology Review 12 (1) (2019) 80 – 86

[14] Mirkovic, J., & Reiher, P. (2004). DoSDefense Mechanisms. 34(2), 39–54. http://delivery.acm.org/10.1145/1000000/997156/p39-mirkovic.pdf?ip=150.183.226.91&id=997156&acc=ACTIVE SERVICE&key=336BF258277217C3.336BF258277217C3.4D4702B0C3E38B35.4D4702B0C3E38B35&__acm__=1519027992_7bc0bb359ba5bc79f940b61d45f461bc.

[15] Mustafa A. and Suraiya P. (2016), "Analysis of Dos and DDos Attacks" International Journal of Emerging Research in Management &Technology ISSN: 2278-9359 (Volume-5, Issue-5) Department of Computer Science, Jamia Hamdard, New Delhi, India

[16] Qasim, B., & Al-Musawi, M. (2012). MITIGATING DoS/DDoS ATTACKS USING IPTABLES. International Journal of Engineering & Technology IJET-IJENS, 12(03), 1210803–1217474.

[17] Ramkumar B. N. & Subbulakshmi T. (2021), TCP SYN flood attack detection and prevention system using adaptive thresholding method. School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, India. ITM Web of conferences.

[18] Rehman, R., & Rahman, O. U. R. (2010). Testing and Analysis of Personal Firewalls.

[19] Russel, R. (2002), Linux 2.4 Packet Filtering HOWTO. http://www.netfilter.org/documentation/HOWTO/packet-filtering-HOWTO.html

[20] Sara A. B. (2018), Towards Securing Web Server Using IptablesUniversiti Sultan ZainalAbidin, Terengganu, Malaysia

[21] Šimon M., HurajL. andČerňanský M. (2015). Performance Evaluations of IPTables Firewall Solutions under DDoS attacks, *11*(2), University of SS. Cyril and Methodius, Trnava. JAMSI, 11 (2015), No. 2 35–45.

[22] Waziri, I. M. (2016). Packet filter performance monitor (anti-DDOS algorithm for hybrid topologies).

[23] Wenhui Su, Junjie Xu (2013), "Performance Evaluations of Cisco ASA and Linux iptables Firewall Solutions" Master Thesis in Computer Network Engineering, School of Information Science, Computer and Electrical Engineering Halmstad University, Sweden

[24] Wu, Q. (2012). The Research and Application of Firewall based on Netfilter. Physics Procedia, *25*, 1231–1235.https://doi.org/10.1016/j.phpro.2012.03.225

[25] Wu, Q., Shiva, S., Roy, S., Ellis, C., & Datla, V. (2010). On modeling and simulation of game theory-based defense mechanisms against DoS and DDoS attacks On Modeling and Simulation of Game Theory-based Defense Mechanisms against DoS and DDoS Attacks, (January). https://doi.org/10.1145/1878537.1878703.