Analysis of UDP DDoS Flood Cyber Attack and Defense Mechanisms on Web Server with Linux Ubuntu 13

Samad S. Kolahi, Kiattikul Treseangrat, Bahman Sarrafpour,
Department of Computing, Unitec Institute of Technology, Auckland, New Zealand
skolahi@unitec.ac.nz
re_vision@live.com
bsarrafpour@unitec.ac.nz

***Abstract*—**Denial of Service (DoS) attacks is one of the major threats and among the hardest security problems in the Internet world. Of particular concern are Distributed Denial of Service (DDoS) attacks, whose impact can be proportionally severe. With little or no advance warning, an attacker can easily exhaust the computing resources of its victim within a short period of time. In this paper, we study the impact of a UDP flood attack on TCP throughput, round-trip time, and CPU utilization for a Web Server with the new generation of Linux platform, Linux Ubuntu 13. This paper also evaluates the impact of various defense mechanisms, including Access Control Lists (ACLs), Threshold Limit, Reverse Path Forwarding (IP Verify), and Network Load Balancing. Threshold Limit is found to be the most effective defense.

1. INTRODUCTION

A revolution has occurred in the world of computer and communication with the advent of the Internet. The Internet has become increasingly important to current society; it has changed our way of communication, business models, and made information publicly accessible quickly and easily from almost anywhere, anytime.

However, with all the advantages of the Internet, there are also some disadvantages. There is no absolute security in the Internet world, and the hackers can use the Internet to launch different types of attacks on a victim network, one of which is known as a Distributed Denial of Service (DDoS) attack.

A DDoS Attack is one of the most common and major threat to the Internet in which the goal of the attacker is to consume computer resources of the victim, usually by using many computers to send a high volume of seemingly legitimate traffic requesting some services from the victim. As a result, it creates network congestion on the target, thus disrupting its normal Internet operation [1].

In particular, a UDP flood attack occurs when an attacker crafts numerous packets to random destination ports on the victim's computer. The victim system, on receipt of the UDP packet requests, would respond with appropriate ICMP packets, if the port is closed [2]. A very large number of packet responses would slow down the system or crash. In this paper, we evaluate the impact of a UDP flood attack on the Web Server with the new generation of Linux platform, namely, Linux Ubuntu 13. This paper also evaluates the existing defense mechanisms such as Access Control Lists (ACL) [3], Threshold Limit [4], IP Verify [5], and Network Load Balancing [6].

ACLs stop the attack by blocking all private IP addresses since they cannot be used on the Internet. Threshold Limit stops the attack by limiting the traffic rate up to the threshold. In this study, we limited the traffic up to 10000 packets per second. IP Verify gives the ability to the router to verify the reachability of the source IP addresses before they can enter the network. If the source IP address is not valid, the packet is dropped. Network Load Balancing can reduce the impact of the attack by balancing the attack traffic to an additional server using different paths and cables.

The organization of this paper is as follows. In the next section, the motivation of this paper and the related work on DDoS Attacks is discussed. Section 3 covers the experimental setup and hardware specification. Section 4 covers information regarding the traffic measurement and data generating tools. Section 5 covers the evaluation of a UDP flood attack and defenses, and the last sections include the conclusions and future works.

2. RELATED WORK

Analysis and comparison of DDoS Attack and defense mechanisms on different operating systems has been conducted by a number of researchers.

In 2006, Pack and colleagues [7] investigated the efficiency of Access Control List against the DDoS Attack. The result shows that the number of ACL rules affects the collateral damage (legitimate traffic was dropped unintentionally). With 5 ACL rules, the number of the collateral damage was 45%. However, this number significantly reduced to 15% if 50 ACL rules were used.

In 2009, Lu and colleagues [8] investigated the impact a UDP flood attack on the system by using metrics such as packet loss rate, delay, and jitter. The testbed consists of 9 routers and 14 computers with Intel Celeron 2.1 and 512 memory running Linux. Iperf was a primary tool used to generate UDP traffic at 10, 15, 20 and 30Mbps. The result shows that without the attack there was no packet loss and the delay jitter value was 32.3%. During a UDP flood attack, however, the number of packet loss went up to 14.08% while the jitter slightly decreased to 29.7%.

In 2009, Rui and colleagues [9] conducted a study of DDoS prevention based on IP Verify and Threshold Limit. The simulation program in this study was .net 2005 running on Windows Server 2003 system and the total number of IP addresses tested was 12,960,000 IP addresses.

In 2011, Subramani [4] conducted an experiment on TCP and a UDP flood attack and proposed 2 defense mechanisms namely Access Control Lists and Threshold Limit. The results show that without the attack, the average response time of the server was 0.834 milliseconds while during the attack this number increased to 8.782 milliseconds. After using Access Control Lists, the average response time went down to 1.093 milliseconds, and it reduced to 6.985 milliseconds when using Threshold Limit.

In 2012, Kaur and colleagues [10] conducted an experiment on DDoS Attack using a DETER testbed. The network in this experiment consisted of three computers: an attacker computer, legitimate computer, and FTP server. The purpose of this research was to study the impact of the user throughput between computer nodes during a UDP flood attack. Traffic result shows that the average bandwidth before the attack was around 75Kbps while during the attacks, the average bandwidth has raised around 130Kbps.

There has been no work done on testing performance and defense mechanisms on Web Server with Linux Ubuntu 13. The lack of available research on the impact of a DDoS attack on Web Server with new generation of Linux platform was the main motivation behind this paper.

3. EXPERIMENT SETUP

The test-bed diagram for site to site is displayed in Figure 1. The test-bed hardware setup remained constant for all experiments conducted.

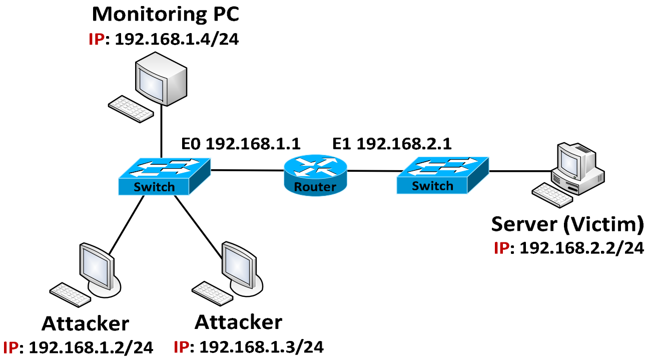

Figure 1: Network test-bed.

The network was setup through a direct connection using a standard category 5e cabling between workstations. The router was used to separate two networks, and used to monitor incoming and outgoing traffic between networks. There were four types of workstations in the test-bed: Two workstations will act as attackers, one will act as a victim, and another one is used as a monitoring machine.

The workstations where the attackers perform have Hping3 [13] as an attacker generator, which is a built-in tool that is offered with Linux Back Track R3. The victim machine with Web Server has Linux Ubuntu 13 installed. The monitoring PC in which Windows 8 installed is where the different varieties of monitoring tools installed to gather data and perform the network testing analysis.

The hardware benchmark comprised of an Intel® Core™ i5 2.80 GHz processor with 8.00 GB RAM for the efficient operation of operating systems, Cisco 2811 and Cisco SG 200 were chosen as the network connection devices.

4. DATA GENERATION AND TRAFFIC MEASUREMENT TOOL

TCPing [11] was the primary tool used to investigate the latency of the web server during the attack. Latency is a measure of time delay experienced in a system. By using TCPing we can measure the response times and hence we have calculated the latency of the victim computer.

Iperf [12] was selected as the tool to measure the user throughput and packet loss during the attack. Iperf has a client and server functionality, and can measure the throughput between the two ends, either unidirectional or bi-directionally. It is open source software and runs on various platforms including Linux, and Windows.

Hping3 [13] was chosen as an attacker generator, which is a built-in tool that is offered with Linux Back Track R3. HPing3 allows users to generate different types of DDoS attacks including UDP, TCP, and Smurf attack.

Webstress Server Tool [16] was used to generate legitimate traffic. It is software for load and performance testing of a webserver. Webstress Server Tool is designed to simulate multiple users accessing to a website.

All performance evaluation tests were run for 5 minutes, which generated the attack traffic at approximately 3.5 million packets per run. The legitimate traffic was generated by Webstress Server Tool, which generate the connection request from users to the webserver assuming on average 10 users per second. To ensure high data accuracy, each test was repeated at least 30 times and data average and runs continued until standard deviation of results was below 0.07% of the average.

5. EXPERIMENTAL RESULTS

The experiments were conducted to evaluate and compare TCP throughputs, round-trip time and CPU utilization before and during the attack on Web Server with Linux Ubuntu 13. This section also evaluates four defense mechanisms, namely, Access Control Lists, Threshold Limit, IP Verify, and Network Load Balancing.

5.1 Impact of a UDP flood attack on Linux Ubuntu 13

Figure 2 illustrates the TCP throughput values before and during the DDoS attack. TCP is the transmission protocols used for webserver traffic.

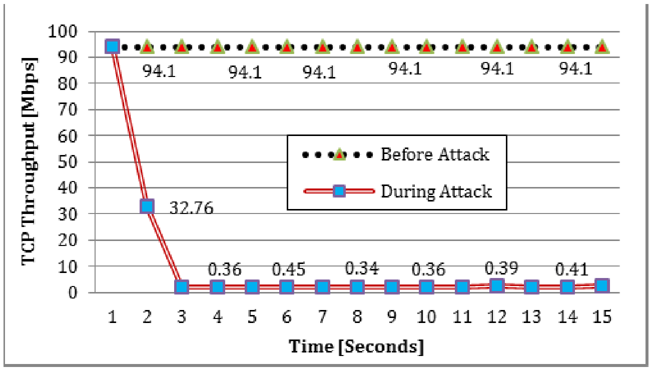

Figure 2: TCP Throughput before and during attack on Web Server.

Overall, the result shows TCP throughput value decreased significantly after launching the attack. The TCP throughput value before the attack was constant at 94 Mbps. During the attack, this figure dropped significantly to 0.36 Mbps. This is because an attacker sent a large number of UDP packets to the victim computer and eventually caused the network congestion. Also, it is caused by the nature of TCP that connection-oriented as it reduces its transmission rate when bandwidth exceeds its receiving ability [16]. It can be noted that throughput value during the attack was almost stable after 3 seconds, and it was between 0.36 to 0.45 Mbps.

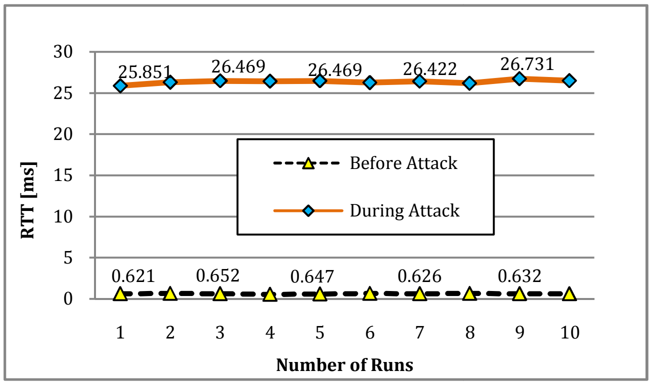

Figure 3: Round-trip time before and during the attack on Web Server.

Figure 3 shows the average round-trip time (RTT) for 30 runs before and during a UDP flood attack on Web Server with Linux Ubuntu 13. The result shows that the average RTT before the attack was 0.62ms. During the attack, however, the average RTT increased significantly to 26.42ms

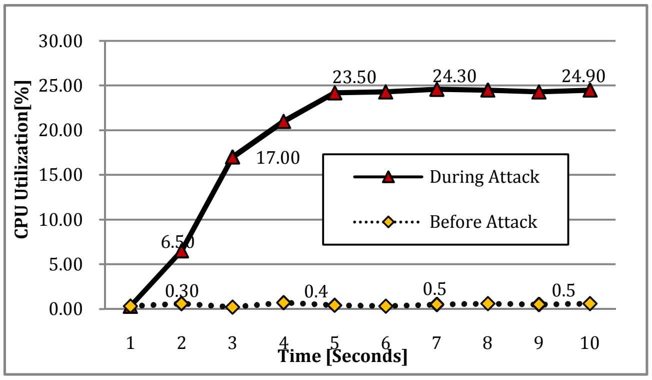

Figure 4: CPU utilization before and during the attack on Web Server.

Figure 4 illustrates the CPU utilization before and during the attack. The result shows that the CPU usage before the attack was constant at 0.3% to 0.5%. During the attack, however, the CPU utilization went up at approximately 6% within 2 seconds and increased to 17% in 4 seconds. Afterwards, it remained steady at 23.5% to 24.9%. The computer needs more processing to action the DDoS UDP flood attack traffic.

5.2 Evaluation of DDoS defenses on Web Server with Linux Ubuntu 13

Figure 5 shows the impact of the UDP flood attack on TCP throughputs after using defenses, namely, ACLs, Threshold Limit, IP Verify, and Network Load Balancing. Additional server was added to do the load balancing.

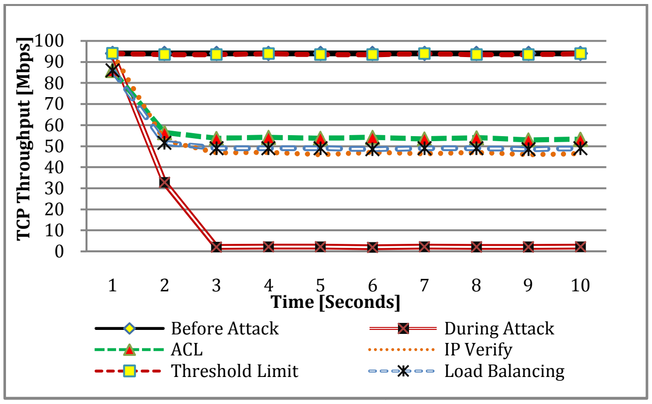

Figure 5: TCP Throughput on Web Server after using DDoS defenses.

Figure 5 shows TCP throughput values were increased after using defenses. The TCP throughput value before the attack was stable at 94.1 Mbps. During the attack, this number significantly dropped to 0.36 Mbps.

In terms of defenses, Threshold Limit [4] was the most effective solution in this study, which increased the throughput value from 0.36 to 94 Mbps. Access Control Lists [3], Network Load Balancing [6], and IP Verify [5] increased moderately the TCP throughput values to 53.37 Mbps, 47.29 Mbps, and 46.93 Mbps, respectively.

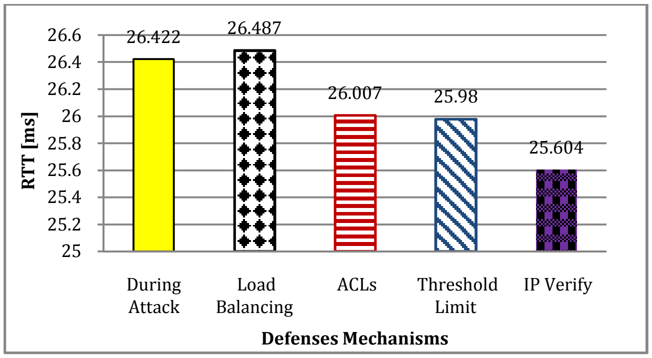

Figure 6: RTT on Web Server after using DDoS defenses.

Figure 6 shows the comparison of round-trip time after using DDoS defenses on Web Server with Linux Ubuntu 13. On the whole, the result shows that the RTT value was decreased after using defenses with the exceptions for Network Load Balancing. IP Verify was the most effective defense in this study, which reduced the RTT from 26.422ms to 25.604ms. Threshold Limit came in second, and reduced the RTT to 25.98ms, while ACLs reduced the delay value to 26.007ms.

Interestingly, Network Load Balancing resulted in the highest RTT, which was 26.487ms. It can be noted that this number was even higher than the RTT value during the attack. The reason is that this defense requires the system resources to examine incoming packets and make load-balancing decisions, and thus impose an overhead on network performance [6].

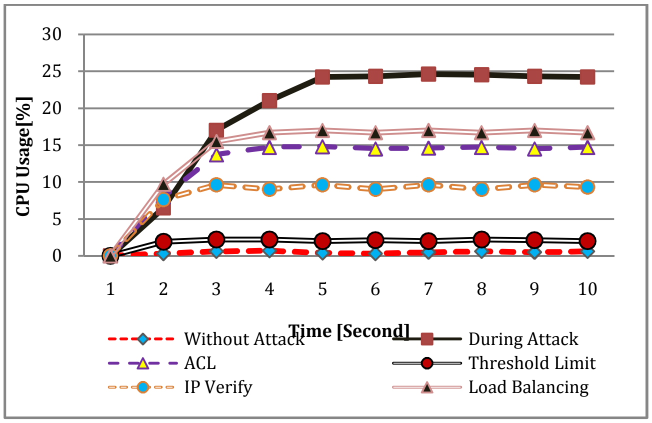

Figure 7: CPU utilization on Web Server after using DDoS defenses.

Figure 7 shows the average CPU utilization before and after using DDoS defenses on Web Server with Linux Ubuntu 13. The result shows that the CPU usage before the attack was stable at around 0.3% to 0.5%. During the attack, the CPU usage increased to 24.9%. The most effective defense in this study was the Threshold Limit, which reduced the CPU usage from 24.9% to 3%. IP Verify reduced the server's CPU to 10%. ACLs reduced the CPU usage from 24% to 15%, while Network Load Balancing decreased the CPU usage to 18%.

6. CONCLUSION

In this paper, we studied the impact of a UDP flood attack on Web Server with the new generation of Linux platform, namely, Linux Ubuntu 13. The result showed that the TCP throughput before the attack was constant at 94.1 Mbps and dropped significantly to 0.36 during the attack. The RTT result shows the average RTT before the attack was 0.62ms. During the attack, the average RTT increased significantly to 26.42ms. The CPU result shows that the CPU usage before the attack was constant at 0.3% to 0.5%. During the attack, the CPU utilization went up to 24%.

After using defenses, Threshold Limit was the most effective defense, which significantly increased the TCP throughput value from 0.36 to 94 Mbps, whereas the rest of defenses could increase the throughputs between 46.93 and 53.37 Mbps. The RTT result after using defenses showed that IP Verify was the most effective defense as it reduced the RTT value from 26.42ms to 25.60ms. On the other hand, Network Load Balancing resulted in the highest RTT, which increased the RTT from 26.422ms to 26.487ms. After using defenses, Threshold Limit significantly reduced the CPU usage from 24.9 to 3% whereas the rest of defenses reduced the CPU usage ranging from 10% to 18%.

FUTURE WORKS

In future, we plan to extend this study by incorporating latest operating systems such as OS X Lion 10.7, Linux Fedora 20, Linux GNOME 3.12, and Windows 8.1 system. In addition, the denial of service attack exploiting IPv6 mobility will be investigated.

REFERENCES

[1] B. Gupta, C. Joshi, and M. Misra. "Distributed Denial of Service Prevention Techniques" IJCEE, vol. 2, no. 3, 2010, pp. 268-276.

[2] A. Singh and D, Junefa. "Agent Based Preventive Measure for UDP Flood Attack in DDoS Attacks" IJEST, vol.2, no. 8, 2010, pp. 3405-3411.

[3] Y. Rekhter. Address Allocation for Private Internets. RFC 1918, February 1996.

[4] R. Subramani, "Denial of Service Attacks and Mitigation Techniques: Real Time Implementation with Detailed Analysis," The SANS Institute, 2011.

[5] F. Baker, and P. Savola. "Ingress Filtering for Multihomed Networks" RFC 3704, March 2004.

[6] Microsoft (2014). Network Load Balancing Technical Overview. Available:http://technet.microsoft.com/en-us/library/bb742455.aspx

[7] G. Pack, J. Yoon, E. Collins, and C. Estan. "On Filtering of DDoS Attacks Based on Source Address Prefixes.," presented at the Securecomm and Workshops, Baltimore, 2006.

[8] W. Lu, W. Gu, and S. Yu. "One-Way Queuing Delay Measurement and Its Application on Detecting DDoS Attack," Journal of Network and Computer Applications, vol.32, no.2, 2009, pp. 367-376.

[9] X. Rui, M. Li, and Z. Ling. "Defending against UDP Flooding by Negative Selection Algorithm Based on Eigenvalue Sets," in International Conference on Information Assurance and Security, Xi'an 2009, pp. 342-345.

[10] D. Kaur, M. Sachdeva. K. Kumar. "Study of DDoS Attacks Using DETER Testbed," in International Journal of Computing and Business Research, vol.3, no.2, 2012, pp. 1-13.

[11] E. Fulkerson. (2014). Ping Over a TCP Connection. Available: http://www.elifulkerson.com/projects/tcping.php

[12] R. Jones. Netperf 2.4.5. Available: http://www.netperf.org/netperf/NetperfNew.html

[13] S. Sanfilippo. (2014). Hping3. Available: http://www.hping.org/hping3.html

[14] S.S. Kolahi, Y. Cao, and H. Chen, "Bandwidth IPSec Security Trade-off in IPv4 and IPv6 in Windows 7 Environment " in Second International Conference on Future Generation Communication Technology (IEEE FGCT) London, 2013, pp. 148-152.

[15] G. Mohd and H. Rosilah, "Flooding Distributed Denial of Service Attacks-A Review," Journal of Computer Science.vol.7, no.8, 2011 pp. 1218-1223.

[16] Webstress (2014). Website Performance, Stress, and Load Testing. Available: http://www.paessler.com/webstress