International Conference on Information and Communication Technology and Its Applications (ICTA 2016)

Federal University of Technology, Minna, Nigeria

November 28 – 30, 2016

Development of a Traffic Analyzer for the Detection of DDoS Attack Source

Joseph Adebayo Ojeniyi[1], Maruf Olalekan Balogun[2], Fasola Sanjo[3], and Onwudebelu Ugochukwu[4]

[1,2]Department of Cyber Security Science, Federal University Technology, Minna, Nigeria

[3]Department of Computer Science, University of Ibadan, Ibadan, Nigeria

[4]Department of Computer Science, Federal University Ndufu-Alike Ikwo, Abakaliki, Nigeria

[1]ojeniyija@futminna.edu.ng, [2]marufbyte@gmail.com, [3]sanjo@elsmedia.com, [4]anelectugocy@yahoo.com

***Abstract*— Distributed Denial of Service (DDoS) attack has been the most devastating attack on computer network and internet at large. Several techniques have been deployed to mitigate this attack. However, detecting the source of DDoS attack remains unsolved in the literature. The aim of this paper is to develop a traffic analyzer for the detection of DDoS attack source. The approach used consists of sniffing, analysis and isolation of source and destination IP address with their respective timestamp of packets that flow through the network in which system was deployed. Traffic analyzer has the ability of saving the captured packet for possible examination and analysis by forensic expert. Traffic Analyzer was developed as a console based application using python programming language which is limited to run on Linux distribution. A network was simulated using GNS3 consisting of the attacker and the victim machine (both run on kali Linux). The result of this work was shown after the developed traffic analyzer was used to collect traffic from the simulated victim machine, thereby showing the traffic and their header information. The arrival time of each IP address that comes inside the network was logged. With this the analyzer was used to determine the type and source of DDoS attack.**

Keywords-network attack, DoS, DDoS, traffic analyzer, detection log, python programming language

I. INTRODUCTION

From the beginning of 21st century, there have been an evolving threat toour cyberspace, these attacks are classified majorly as attack against confidentiality, integrity and availability of information. Distributed Denialof Service (DDoS) attack have been the most devastating attack on our network and internet at large and they are being tagged as the attack against availability of information whereby the information that are meant to be available for a legitimate user is being denied by the server because the attacker is accessing the server and sending unsolicited request to this machine thereby causing the legitimate client inability to access its resources. A Distributed Denial of Service (DDoS) is where the source of attack is more than one and often thousands of unique or spoof IP addresses. Perpetrators of DDoS attacks often target sites or services hosted on high-profile web servers such as banks, credit card payment gateways, but motives of revenge, blackmail or hacktivism can be behind other attack like the attacks reputable companies or countries.

II. LITERATURE REVIEW

A. Related Work

At the present time, more and more critical infrastructures arebeing used by organizations and they are increasingly relying upon the internet in order to carry out their day to day operations [1]. Internet attacks are at increasing rate and threat are also increasing to cripple Information Technology infrastructures [2].

With the increase in large attacks that directly targets the large businesses and government institutions around the world, one of the most significant issues that can be considered by both commercial and governmental organizations is to protect its information from malicious jeopardizing that is, the adoption network security is more important now more than ever because of the increase in attacks every day by day due to the automated tools being use against internet-connected systems by attackers [1]–[4].

Denial of service (DoS) or Distributed Denial of Service (DDoS) attacks is one of the most devastated internet attack against internet connected system in this era and they can be defined as attempts to make a computing or network resource unavailable to its users or as an attack that pose a highly damageable threat to the CIA (Confidentiality, Integrity and Availability) of services that resides on the network [4]–[6]. DoS attack often involve using a single computer in preventing the legitimate users from accessing the network resources while the advance DoS attack which is Distributed Denial of Service (DDoS) attack involves multiple compromised computer being used to send attacks to a victim at the same period during the attacking time [4], [5]. DDoS attack is mainly achieved with the help of botnet which are refers to as compromised systems under the instructions of their master or handlers [7]. Botnet can also be refer to as zombie and they are responsible for generating the attack traffic towards the victim [8].

Basically, DDoS attack architecture consists of three components which are master, slave and the victim. They collaboratively work together towards achieving their malicious goals. Figure l shows the model of a typical DDoS attack. The master takes control of the botnet without the knowledge of their owners, because they have been previously infected with a Trojan or a backdoor program. The compromised machines called botnet are being control by the bot-master, often through Command and Control (C&C) channels, and simultaneously used to track a victim using the public internet infrastructure [9].

Internet crime like DDoS attack is still at large and on the rise there is not yet an effective and efficient system to know where the malicious packet come from, or where the suspect is located so that he/she can be identify, track, report, arrest and punish for its offence [1].

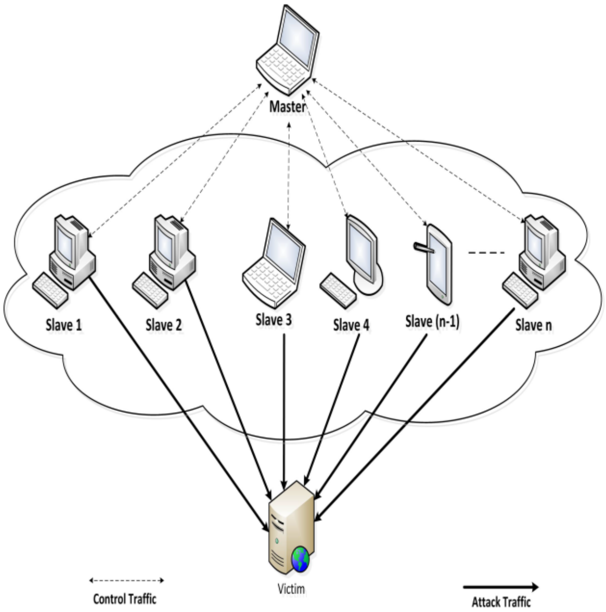


Figure 1. Showing the typical set-up of a DDoS attack

B. *Summary of Review*

The summary of the review based on this research work is shown below in Table 1.

III. SYSTEM DESIGN

In order to have detailed understanding about the proposed system. This section explains functions of the proposed system using system flowchart and UML Use Case Diagram

A. *System flowchart*

The propose traffic analyzer for detection of DDoS attack source flowchart is depicted below in figure 2. Because of the sniffing and reporting features of this system, running the system will enable it to start capturing packet from the Ethernet frame either through wireless or wired network. This packet would contain the following relevant information:

- Source and Destination IP address
- Source and Destination MAC address
- Source and Destination Port number

After the collection of this information, it will store the destination and source IP address of every new packet and then check if the source and destination is existing inside the hash table which can be refer to as dictionary in python. If the information is existing, it will add the occurrence of this IP addresses into their respective hash table for further network traffic analysis.

B. *UML Use Case Diagram*

This section uses the UML use case diagram to explain the proposed system. Use case diagram has been known to consist of mainly the actors and their respective functions. Figure 3 depict the proposed system in which the actor is represented by as proposed system with its respective features which to sniff packet, analyze traffic and report engine.

The sniff packet use case represents the ability of the system to be able to capture packet that comes in and out of the network computer putting the following criteria into consideration.

- Source and Destination IP address
- Source and Destination MAC address
- Source and Destination Port number

The network traffic analysis component would check for the following information which can be useful in case if there is an existence of DDoS attack in that particular network.

- The packet protocol
- The packet header

The report engine consists of two functions which are logging evidence creation after the system termination. The following information are logged in order to examine the existence of DDoS attack.

- Source and Destination IP address
- Their respective timestamp

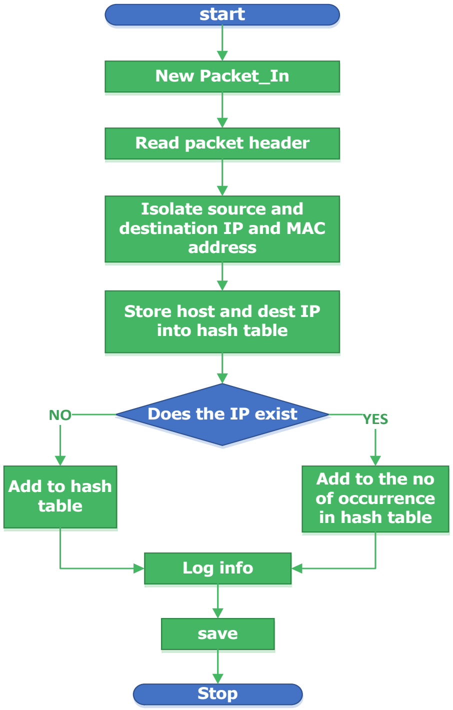

start
New Packet_In
Read packet header
Isolate source and destination IP and MAC address
Store host and dest IP into hash table
Does the IP exist
NO
YES
Add to hash table
Add to the no of occurrence in hash table
Log info
save
Stop

Figure 2. Attack source flowchart

TABLE I. SUMMARY OF RELATED WORKS

Author/Year	Methodology	Achievement	Limitation
[10]	Using entropy based algorithm	Entropy network based anomaly detection method	Limited to solving single label problem
[11]	RBPBoost was trained and tested with DARPA, CONFICKER,	Improving on RBPBoost Algorithm	Limited to known attack detection
[12]	IBR analyzer using python	Able to develop a system for analyzing capture data from IBR	Limited to characterizing IBR information to their respective payloads
[13]	Data analysis and reporting tool.	Ability to receive a .pcap file and transform it into report format	Limited to processing smaller packets
[14]	Modelling and Countermeasures Using Botnet and Honeypots	An information-theoretic framework models for flooding attacks using Botnet on ITM and effective attack detection using Honeypots	Limited to small and homogenous network
[4]	A data mining Centroid-based rule method	DDoS attack Detection and defense approach	Stability of centroid-based rules for non-spherical shapes
[2]	Virtual honeynet data collection mechanism	Detection of IRC and HTTP botnet	Focusing on botnet detection on network –level traces
[9]	Using ensemble-based DDoS attack detection and rate of change of unseen IP addresses	DDoS detection techniques	Limited to equal weight simple correlation
[15]	Greedy layer wise unsupervised training strategy	Training deep neural network for DDoS detection	Techniques works for unsupervised training only
[16]	Valuation method of probability loss of arbitrary request passing on mass network service	Detection technique for DoS/DDoS/DRDoS attacks in network mass service	Only applicable in stationary mode
[17]	A statistical CUSUM-based detection technique	Detection of DDoS attack	Technique depends on CUSUM
[8]	Entropy based algorithm	Early detection of DDoS attack in software defined network	Limited to detection of attack when the DDoS attack is targeting a host not the entire network
[18]	Combining multiple independent data sources to study large DDoS attacks	A measurement study for analyzing DDoS attack for multiple data sources.	
[19]	Using PMD technique and labelling of incoming packet in detection of sniffing and DDoS attack	Detection and isolation of DDoS attack with packet sniffing in a SCADA network	The both techniques work separately in detecting their target
[1]	Flexible Deterministic Packet Marking technique	An IP traceback system that is having high probability of finding the source of DDoS attack	Processing of packet consume more resources
[20]	Flexible Deterministic Packet Marking technique	An IP traceback system that is having high probability of finding the source of DDoS attack	It requires human intervention. i.e. it is not automated. May not be able to give high performance in a large network

The timestamp would be logged first followed by the respective IP address in order to map host IP address with their respective source address of every flow of packet in and out of the network in order to ease the investigation of potential DDoS attack source and where the attack is really targeting.

IV. . METHODOLOGY

A. System Requirement

In order to achieve this project, there are the requirement that must be met for both the software that would be used in building the system and the hardware specification needed to simulate the DDoS embedded network.

1) Hardware and Software requirement

- GNS3 software for the DDoS embedded network simulation
- Kali linux operating system (for both and victim and attacker’s machine).
- Virtual machine (Virtual Box or VMware)
- The specification that would be needed in other to perform achieve this project is minimum of 500GB hard drive, 8GB RAM and 2.35GHz quad core laptop

2) Tools and libraries needed:

a) Python programming language: Python was chosen over the other programming languages because python is beginner’s friendly and the choice of language penetration testers and forensic analyst and entire cyber security field at large.

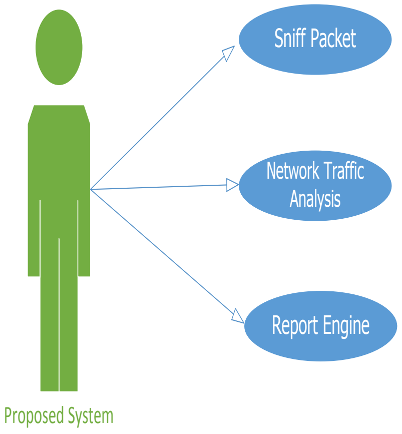


Figure 3. use case diagram of traffic analyzer

b) Socket: This module offers access to the socket interface of BSD and is available on all current Unix systems, Windows, MacOS, and possibly additional platforms. This module provides everything you need to build socket servers and clients.

c) Struct library: It does changes between Python values and C structs characterized as Python bytes objects which are use to handle binary data stored in files or from network connections, amid other sources. Format Strings is use as solid descriptions of the C structs plan and the intended change to/from Python values.

d) Datetime library: This module is responsible for providing classes in order to manipulate dates and times in both simple and multipart ways. While date and time arithmetic is maintained in this module, the motivation of this application is to efficiently extract attribute for output formatting and manipulation

e) Time library: It provides various time-related functions. Almost all the functions defined in this module call platform C library functions with the similar name.

f) Textwrap: It is one of the module that perform text processing services. This module provides the functions of wrapping or filling one or two text strings. It also has some convenience functions, as well as Textwraper, the class that does all the work. Textwrap is would be use in the formatting and arrangement of string.

V. IMPLEMENTATION AND RESULTS

A. Introduction

This section reports the implementation of the developed system (Traffic Analyzer), and also Distributed Denialof Service (DDoS) attack network simulation which was used as the test bed in order to carry out the system testing of the developed system.

B. Tools needed for system implementation

Below are the tools that are used in achieving our DDoS test bed in order to further test the workability of our developed system.

1) Graphical Network Simulator3 (GNS3)

GNS3 is a network simulator that allows simulation of networks. It consist of Dynamips (a cisco router emulator) and also contains Pemu (a cisco PIX firewall emulator) as well as tight incorporation with wireshark (packet capture and protocol analyzer).

2) Hping

hping is a command-line oriented TCP/IP packet analyzer/assembler. It supports TCP, UDP, ICMP and RAW-IP protocols, has a traceroute mode, the capability to send files between an enclosed channel, and many other features. one of the features of hping command is network testing, this network testing feature was use to perform DDoS attack against the victim's machine.

C. DDoS Test Bed to test Traffic Analyzer

The system testing environment was achieved by simulating a network which was in carrying out Distributed Denialof Service attack of the attacker's machine while the develop system is set up on the victim's machine. Figure 4 shows how this was achieved using Graphical Network Simulator (GNS3).

The router shown in this figure 4 was configured in order to connect the two-dissimilar network of /24 netmask, the attacker's network (192.168.1.0/24) and the victim's network (10.10.0.0/24). The Kali linux operating system to act as our attacker and victim in our test bed.

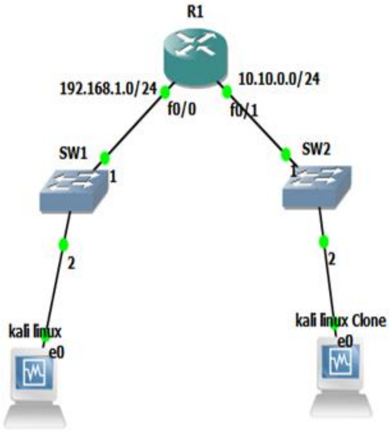
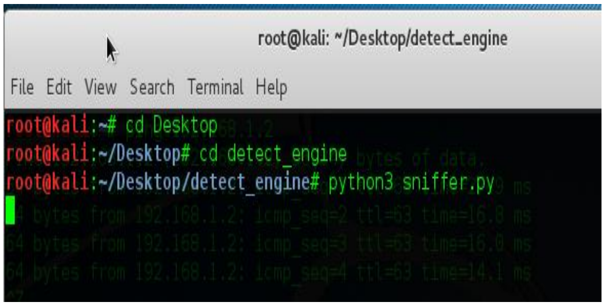


Figure 4. DDoS test bed for the system testing

D. System Testing and Result

The developed system was implemented on the Kali linux clone because is the system that was configured to act as the victim machine while the Kali linux at the left-hand side of figure 4 was configured to act as our attacker machine. Figure 5 shows the implementation of our developed system using python programming language version 3. Both Kali linux and Kali linux clone are assigned IP address of 192.168.1.2 and 10.10.1.3 respectively.

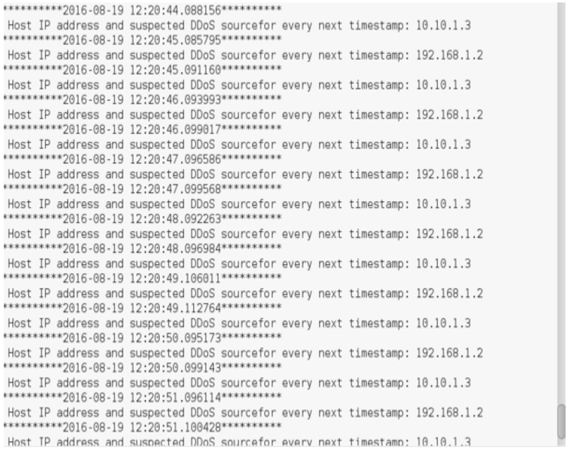
When this developed system is run on any system, it turns the system network interface card (NIC) into promiscuous mode then it begin to sniff every that comes in and out of that network the system is connected to, analysis the traffic and log IP addresses information by mapping the source and destination IP address with their timestamp and finally save all the capture packet in a pcap file format.



```
root@kali:~# cd Desktop
root@kali:~/Desktop# cd detect_engine
root@kali:~/Desktop/detect_engine# python3 sniffer.py
```

root@kali: ~/Desktop/detect_engine
File Edit View Search Terminal Help

Figure 5. Implementation of Traffic Analyser



```
*********2016-08-19 12:20:44.088156*********
 Host IP address and suspected DDoS sourcefor every next timestamp: 10.10.1.3
*********2016-08-19 12:20:45.085795*********
 Host IP address and suspected DDoS sourcefor every next timestamp: 192.168.1.2
*********2016-08-19 12:20:45.091160*********
 Host IP address and suspected DDoS sourcefor every next timestamp: 10.10.1.3
*********2016-08-19 12:20:46.093993*********
 Host IP address and suspected DDoS sourcefor every next timestamp: 192.168.1.2
*********2016-08-19 12:20:46.099017*********
 Host IP address and suspected DDoS sourcefor every next timestamp: 10.10.1.3
*********2016-08-19 12:20:47.096586*********
 Host IP address and suspected DDoS sourcefor every next timestamp: 192.168.1.2
*********2016-08-19 12:20:47.099568*********
 Host IP address and suspected DDoS sourcefor every next timestamp: 10.10.1.3
*********2016-08-19 12:20:48.092263*********
 Host IP address and suspected DDoS sourcefor every next timestamp: 192.168.1.2
*********2016-08-19 12:20:48.096984*********
 Host IP address and suspected DDoS sourcefor every next timestamp: 10.10.1.3
*********2016-08-19 12:20:49.106011*********
 Host IP address and suspected DDoS sourcefor every next timestamp: 192.168.1.2
*********2016-08-19 12:20:49.112764*********
 Host IP address and suspected DDoS sourcefor every next timestamp: 10.10.1.3
*********2016-08-19 12:20:50.095173*********
 Host IP address and suspected DDoS sourcefor every next timestamp: 192.168.1.2
*********2016-08-19 12:20:50.099143*********
 Host IP address and suspected DDoS sourcefor every next timestamp: 10.10.1.3
*********2016-08-19 12:20:51.096114*********
 Host IP address and suspected DDoS sourcefor every next timestamp: 192.168.1.2
*********2016-08-19 12:20:51.100428*********
 Host IP address and suspected DDoS sourcefor every next timestamp: 10.10.1.3
```

Figure 6. IP address log

This traffic analyzer was use in analyzing internet control message protocol (ICMP) packet which gives every parameter of ICMP packet with their values and displaying the IPv4 header with their necessary information.

In a situation whereby the network administrator or whoever is responsible of inspecting the system arrive, all that is needed first is to go to the detection log (figure 6) to check for the IP address log and the respectively timestamp.

E. DDoS attack using IP spoofing

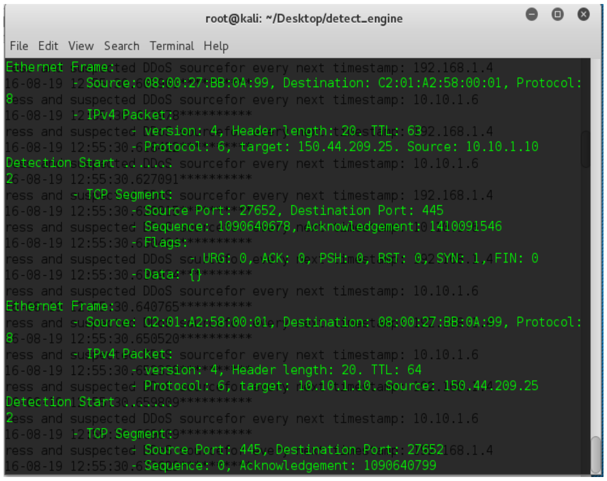
This is experiment the attacker's machine was assigned an IP address of 192.168.1.4 while the victim's machine was assigned an IP address of 10.10.1.10.

Hping command tool is also use in performing DDoS attack using spoofed IP source. This command enables the attacker's machine to send TCP request the victim's machine in which the IP address is spoof in every request that was sent to the victim's machine.

After this command was launched, the traffic analyzer on the victim's machine start capturing packet coming in and analyzing it second Ethernet frame.

Although traffic analyzer was unable detect the real source IP address of the packet but fortunately because most automated software being used to perform DDoS attack do not spoof the attacker's MAC address, it only spoofs their IP address which enable traffic analyser to still a lead of who the attacker's machine is using the MAC address

The detection log shown in figure 7 also shows the logging of the spoof IP addresses with their respective timestamp. looking at the timestamp, that is, how close a request is being sent to the victim before another IP address will make the examiner suspect that the traffic is not a legitimate traffic and it may be DDoS attack. With this, the examiner can then terminate the traffic analyzer to get the save capture file, open it with any pcap reader to check for the MAC address of the suspected IP addresses, all the IP addresses have the same MAC gives proof the evidence that they are all spoof address form a particular source and it is likely to be a DDoS attack.



```
Ethernet Frame:
 - Source: 08:00:27:BB:0A:99, Destination: C2:01:A2:58:00:01, Protocol:
8
 - IPv4 Packet:
     - version: 4, Header length: 20. TTL: 63
     - Protocol: 6, target: 150.44.209.25. Source: 10.10.1.10
Detection Start .......
2
 - TCP Segment:
     - Source Port: 27652, Destination Port: 445
     - Sequence: 1090640678, Acknowledgement: 1410091546
     - Flags:
         - URG: 0, ACK: 0, PSH: 0, RST: 0, SYN: 1, FIN: 0
     - Data: {}
Ethernet Frame:
 - Source: C2:01:A2:58:00:01, Destination: 08:00:27:BB:0A:99, Protocol:
8
 - IPv4 Packet:
     - version: 4, Header length: 20. TTL: 64
     - Protocol: 6, target: 10.10.1.10. Source: 150.44.209.25
Detection Start .......
2
 - TCP Segment:
     - Source Port: 445, Destination Port: 27652
     - Sequence: 0, Acknowledgement: 1090640799
```

root@kali: ~/Desktop/detect_engine
File Edit View Search Terminal Help

Figure 7. Detection log from the DDoS attack

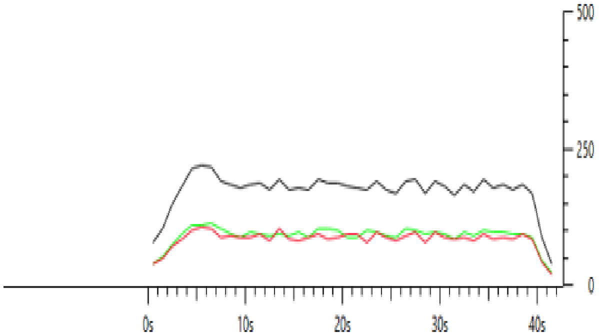


Figure 8. Graph of DDoS

After the DDoS attack was terminated, the save captured packet gotten from this experiment was analyse using statistical (IO Graph) in wireshark to get the graphical presentation of the DDoS attack on the victim machine.

Figure 8 displays the graph of packet sent on the y-axis against x-axis of time 1 second interval. The black line indicate the total total traffic, red line indicate the tcp reset while the green line indicate tcp syn. looking at the figure below we can deduce that the rate of packet that entered the victim machine in the first 40 seconds rises to 250 packet per second and also the tcp syn and tcp reset are almost on the same range. This means that tcp reset by the attacker's machine after every tcp synchonization reset which does not conclude any successful three way handshake. The has proof the traffic is not a legitimate traffic but rather an illegitimate traffic with the characteristics of DDoS attack because IP address are being change after every tcp reset.

This DDoS attack result in the victim's machine unable to respond to even ping request because of the machine resources has been overwhelmed.

VI. CONCLUSION

Development of a traffic analyzer for the detection of Distributed Denial of Service attack has been successfully designed, implemented, tested. This new developed system would help its user to detect anomalous in their production network. It will also help network forensic analyst to easily examine the packet capture from its client network with the help of the save captured packet and detection log features of traffic analyzer. The detection log is always saved as a text file which enables an easy disaster recovery of it, in case if the system crashes, because base on experience, text file is easier to recover compare pcap. Therefore, even though both the captured packet and the detection was lost in an event of disaster, there are still chances of recovery the text file which can also give us some clue what really happened.

REFERENCES

[1] M. S. Asalkar, Sameer A. Bhatnagar, S. Ashish, and A. K. Rahul, “Flexible Determinisic Packet Marking : An IP Traceback System To Find Real Source Of Attack,” 2014.

[2] J. S. Bhatia, R. K. Sehgal, and S. Kumar, “Botnet Command Detection using Virtual Honeynet,” vol. 3, no. 5, pp. 177–189, 2011.

[3] E. M. Angurala and E. M. Rani, “Design and Develop an Intrusion Detection System Using Component Based Software Design,” no. April, pp. 854–860, 2014.

[4] W. Bhaya and M. E. Manaa, “A Proactive DDoS Attack Detection Approach Using Data Mining Cluster Analysis,” vol. 5, no. 4, pp. 36–47, 2014.

[5] Y. Orzach, “Network Forensic with Wireshark,” vol. 3, no. 7, 2014.

[6] K. M. Prasad, A. R. M. Reddy, and K. V Rao, “DoS and DDoS Attacks: Defense, Detection and TracebackMechanisms -A Survey,” *Glob. J. Comput. Sci. Technol.*, vol. 14, no. 7, 2014.

[7] N. E. W. Features, “CISC0 4240 INTRUSI0N PREVENTI0N SENS0R, (2682), l4,” 2004.

[8] S. M. Mousavi, “Early Detection of DDoS Attacks in Software Defined Networks Controller Early Detection 0f DDoS Attacks in Software Defined Networks Controller,” 2014.

[9] S. Bhatia, “Detecting Distributed Denial-of-Service Attacks and Flash Events.”

[10] B. Przemyslaw, J. Bartosz, and S. Marcin, “An Entropy-Based Network Anomaly Detection Method l,” 2015.

[11] M. Kale and D. M. Choudhari, “DDoS Attack Detection Based on an Ensemble of Neural Classifier,” vol. 14, no. 7, pp. 122–129, 2014.

[12] D. Yates, “A System for Characterising Internet Background Radiation,” 2014.

[13] I. van Zyl, “Creating a flexible data processing engine for large packet capture datasets,” 2014.

[14] K. M. Prasad, A. R. M. Reddy, and M. G. Karthik, “Flooding attacks to Internet Threat Monitors (ITM): Modeling and Countermeasures using Botnet and Honeypots,” *J. Comput. Sci.*, vol. 3, no. 6, pp. 159–172, 2011.

[15] H. Larochelle, Y. Bengio, and P. Lamblin, “Exploring Strategies for Training Deep Neural Networks,” vol. 1, pp. 1–40, 2009.

[16] G. Shangytbayeva, G. Kazbekova, U. Imanbekova, S. Munsyzbaeva, and N. Shangytbayev, “Detecti0n Techniques 0f DoS / DDoS / DRDoS Attacks in Networks of Mass Service,” 2015.

[17] M. Alenezi and M. Reed, “Methodologies for detecting DoS/DDoS attacks against network servers,” in *Conference on Systems and Networks*, 2012, pp. 92–98.

[18] Z. M. Mao and O. Spatscheck, “Analyzing large DDoS Attacks Using Multiple Data,” in *International Conference on large-Scale Attack Defense*, 2006, pp. 161–168.

[19] S. Shitharth and D. P. Winston, “A Comparative Analysis between Two Countermeasure Techniques to Detect DDoS with Sniffers in a SCADA Network,” *Procedia Technol.*, vol. 21, pp. l79–l86, 2015.

[20] P. G. Kukreja and D. N. Rewadkar, “Flexible Deterministic Packet Market : An IP Traceback Scheme,” vol. 40, no. 1, p. l595–l60l, 2015.