Securing and Monitoring of Bandwidth Usage in Multi-Agents Denial of Service Environment

Ogunleye G.O.[1]
Department of Computer Science,
Federal University, Oye-Ekiti, Ekiti
State, Nigeria.

Fashoto S.G[2]
Department of Computer Science,
University of Swaziland, Kwaluseni,
Swaziland

Mbunge Elliot[3]
Department of Computer Science,
University of Swaziland, Kwaluseni,
Swaziland

Arekete S.A[4]
Department of Computer Science, Redeemer's University,
Ede, Osun State, Nigeria

Ojewumi T.O.[5]
Department of Computer Science, Redeemer's University,
Ede, Osun State, Nigeria

***Abstract*—The primary purpose of Denial of Service attack (DoS) is to cripple resources so that the resources are made unavailable to the legitimate users. Due to the inadequate monitoring of activities on the network, it has resulted into huge financial losses. Bandwidth which is one of the resources being used on the network, if not properly monitored could result into misused and attack. This paper proposes a real time system for securing and monitoring the amount of bandwidth consumed on the network using the multi-agent framework technology. It also keeps a record of internet protocol (IP) addresses visiting the network and may be used as a starting point for the aspect of response in providing a comprehensive solution to DoS attacks. The bandwidth is pre-entered and an agent is assigned to monitor bandwidth consumption rate against the set threshold. If the bandwidth is consumed above the bandwidth limit and time set, then a DoS attack is suspected taking into considerations the DoS attack framework. This framework can be used as a replicate of what happen in the network scenario environment.**

***Keywords*—Bandwidth; mobile agent; multi-agents; DoS**

I. Introduction

The internet is an interrelated computer networks on a universal system using protocol of the internet (TCP/IP) to connect devices all over the world. Over 4 billion users make use of the internet globally, with an increase of 20% internet users from Africa between 2017 and February 2018 [1,2]. The internet is used for different purposes at different levels to accomplish or support daily activities such as but not limited to research, downloading, electronic mail and group discussion. The over reliance of computer interconnectivity on the internet expose system resources and infrastructure to malicious users. It takes undue advantage of the situation to launch their denial of service (DoS) attacks by interrupting authorized users access which leads to unavailability of computer resources temporarily or indefinitely. The DoS attacks is a malicious attempt that can be achieved in different ways by a person or group of people through computer viruses, worm, Trojan horses, spyware, malware, phishing and so on. Recently, it was discovered that the DoS attacks growth is increasing exponentially on a yearly basis which is a cause of concerns especially on distributed enterprises and small-medium size businesses in 2017 in which 19 million malware attacks was identified and blocked [3,4]. One of the common cyber security threats is the distributed denial of service (DDoS) attack. A DDoS attack is a cyber-attack that takes place when multiple systems bombard the bandwidth or resources of a targeted system with so much traffic to render the services or the infrastructure of the web servers or websites unavailable or useless. The main focus of the attack is to render authorised user incapacitated from being able to perform normal transactions [5,6,7]. Conventional architecture of the internet is vulnerable to distributed denial of service (DDoS) attacks. According to National institute of standard and technology [8,9,10], several vulnerabilities preventive measures should be in place to mitigate malicious attacks but instead vulnerabilities are on the increase. In 2014, 7937 vulnerabilities were recorded as against 5186 in 2013 which shows an increase of 34.6 percent [11,12,13]. From the report of worldwide infrastructure security in 2010 that DoS attack is standardized and becomes scarier to the network operators which make them believe that there will be more problems with DDoS attack (Dobbins and Morales, 2010). The largest and the most impactful DDoS attacks were targeted at GitHub and it occurred on 28th of February 2018 but were reported in March 2018. Irrespective of the shortcomings of the DDoS defense detection mechanism to discover attack close to the target machine, describe the distinctive nature of it, and the inability to filter out legitimate packets from the attack packets, the defense detection accuracy mechanism is still very high. Recently, numerous DDoS defense detection strategies has been proposed such as SDN-based "moving target defense", to secure computer networks and operators from DDoS attacks by moving the computer networks and the operators from targeted virtual machines(VMs) to invulnerable environment [14,15,16].

Despite the high level of DDoS defense detection mechanism accuracy; there is still a need to develop a system capable of detecting and monitoring bandwidth to suspect DDoS attacks on the computer networks. In the fight against DoS attacks some problems have been identified.

The following are some of the attacks facing most networking environments.

- Difficulty in detecting highly distributed Denial of service especially close to its source IP address
- Difficulty in detecting DoS attack immediately without raising false alarm and
- Not delaying response in order to ensure that the attack is not a legal increase in user request on a website.

This paper seems to address the problems by proposing a secured system capable of monitoring the traffics on the network using the mobile agent technology.

II. Literature Review

A. Denial of Service

Denial of Service (DoS) attack occurs when the system denies legitimate users to have access to systems resources [17,18,19]. The DoS attack usually uses the single Internet connection and a single computer to flood a targeted system or resource [20,21,22]. The ultimate goal of DoS is to make services inaccessible [23,24,25] by either injecting computer viruses or flooding network traffic. Flooding network traffic can be achieved by incessant exploiting network vulnerable security loopholes, illegally access to network servers, and then brings down network services [26,27,28]. DoS attack utilises Transmission Control Protocol (TCP) connection buffers, exhausting bandwidth links, network router processing capacity [29], and application layer buffers that can lead to degradation of network performance [30], shun all network connections , and ultimately block the website [31,32]. The most active modern type of DoS is called Distributed Denial of Service (DDoS), which consists of multiple computers and Internet connections to target a single system. With the DDoS attack, incoming network traffic floods victim’s system or resource indirectly using large-scale computer multi-agents connected via the internet [33]. DDoS attacks can happen in two ways over the internet. The first technique is to send the malicious network packets or codes to confuse either security application executing on victim’s machine or Internet Protocol (IP). The second technique is to flood, interrupt and exhaust network connections by using the following three layers of OSI model; network layer, transport level and application. Once the computer is flooded with malicious codes, the multi-agent scans for another vulnerable computer entire the network. Multi-agents search and scan vulnerable computers over the network by employing random scanning, topological scanning, permutation scanning and local subnet scanning. According to Jelena and Peter [34], in random scanning, each compromised computer probes random addresses in either global or local Internet Protocol address space because attacking hosts duplicate themselves and execute uncoordinated attacks, thereby increasing the possibilities of packets collision and high traffic volume due to computers probing same address [35]. Topological scanning utilizes data stored in the victim’s computer to discover new targets. With topological scanning, agents use valid URLs in the Web Servers in victim’s computer to determine the next vulnerable computer in the network. The performance of this technique is almost similar to hit-list scanning [36]. Hit-list scanning occurs when attackers start by scanning the network and gather a list of potentially vulnerable computers before attaching. Once a list is created, attackers incessant rescanning the network to find the vulnerable computers, install malicious code and divide the list into half [37]. A newly infected computed is allocated half of the list, keep the remaining half, and scans the other residual list. When attackers find other vulnerable computers, they apply the same procedure decimating network performance due to the proliferation of infected computers in the network. Hit-list scanning makes sure that all vulnerable computers on the list are infected with malicious code. Local subnet scanning acts behind a firewall in an area that is considered to be infected by the malicious scanning program. The infected hosts scan and target vulnerable computers of its’ own network using information in the local subnet addresses [38]. In permutation scanning, all machines share a common pseudorandom permutation list of Internet Protocol addresses. A block cipher of 32 bits is used to create a permutation list [38]. An infected host by either local subnet scanning or hit-list scanning starts scanning just after its point in the permutation list and scans through this list to find new targets. When it finds a host or a node that is already infected, it starts to scan again at a random point in the permutation list. According to [39], the process of scanning stops when the compromised host encounters sequentially a predefined number of already infected machines without finding new targets during that particular moment.

B. Types of DDoS Attacks

DDoS attacks are typically be grouped as bandwidth and resource depletion attacks. Bandwidth depletion attacks are further sub-divided into two groups namely; flood attacks and amplification attacks. TCP SYN flood, UDP Flood, ICMP Flood and Smurf attacks are examples of bandwidth depletion attacks. With flood attacks, large volumes of malformed packets are sending continuously to the target computers to ensure that buffer overflow occurs and exploit vulnerable hosts. The ultimate goals of flood attacks are; to reduce processing capability, the memory of the victim’s computer, and exhaust packet buffers and network bandwidth over the internet. Flood attacks can be mitigated by software patching. Resource depletion involves zombies sending messages to a broadcast IP address to cause all system in the subnet reached by the broadcast address to send a reply to the victim system [40]. Examples of resources of depletion attacks are; protocol exploits attacks and malformed packets attacks [41]. Figure 1 shows bandwidth depletion and resource depletion attacks. The list is not exhausted since attacks are dynamic and new on daily basis.

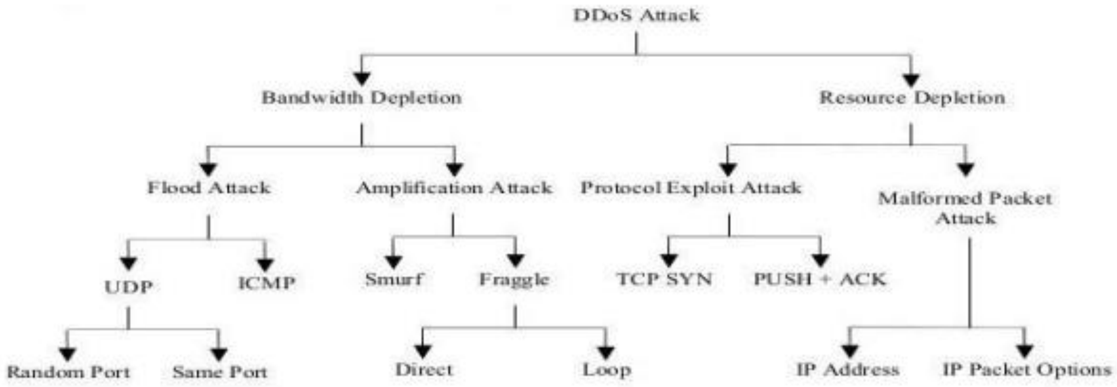


Fig. 1. DDoS Attacks Source (Keyur & Vivek, 2015)

C. DDOS Defence and Detection Techniques

According to [42] a router-based packet filtering, ingress filtering, and source address validation are classified as DDoS prevention methods. Ingress filters depend on addresses that are on router that connects ISP's scope of control to fight against DDoS. Internal interface and external interface acts as a channel for internal addresses and external addresses respectively. These strategies work poorly for transit and backbone hosts but execute perfectly for leaf nodes. Router-based packet filtering and ingress filtering techniques does not affect attackers who show their IP addresses. Router-based packet filtering utilizes internet routing data to whether packets are arrived safely at its designated destination address without any modification [42]. Router-based packet filtering implements fast filtering to determine catch misconfigurations based on router's look-up table. Another effective DDoS prevention technique is called source-address validation. Past authors [42] states that source-address validation relies on building a second routing table of learned networks and matches subnet-to-subnet the 'allowed routes' and the filtered routes. The system changes gradually to topology modifications when the traffic is spoofed. This technique does not consider non-spoofing attacks.

DDoS detection techniques consist of artificial neural networks (ANN), data mining and fuzzy expert systems. Some authors used ANN to detect known and unknown DDoS attacks. ANN consists of the input layer, hidden layer, and output layers layer to make reasonable decisions. The input layer represents the typical patterns of attacks from genuine traffic. Input values in the input layer are feedforward to the preceding layer, hidden layer, which performs some computations by using activation function. The outputs of the hidden layer are used as inputs of the output layer. The learning algorithm and the nature of the problem determine the training function of ANN. Alan, et al.,[43] adopted supervised Backpropagation algorithm to train the network for known DDoS attacks. For unknown DDoS attacks, they trained ANN using unsupervised learning algorithm. Alan, et al, [43] successfully detected forged packets; the defense mechanism was activated to drop the packets while allowing genuine packets to pass through. Blocked packets are unblocked as soon as the system flags the traffic flagged to be normal. Some authors [43] applied Neural Network Classifier to detect a DDoS attack on DNS server. A Neural Network classifier used characteristics of the attacks as input values to classify whether the attacks exist or not. The performance and accuracy of neural network classifier depend largely on whether the selected features can really summarize the characteristics of DDoS attack. Abhilasha and Santosh [44] implemented Artificial Neural Network based on genetic algorithm and multivariate correlation analysis to detect DoS attacks. Support Vector Machine (SVM), data mining and fuzzy expert systems have been used successfully to detect both DoS and DDoS attacks on the internet. Fuzzy expert systems involve a set of rules written using conditional statements to specify attacks into categories and draw some conclusions from facts and rules [44]. All incoming network traffic instances are compared with rules in the system to check whether conditions are satisfied, if not then the system can detect some anomalies and inconsistent in traffic instances.

DoS and DDoS attacks can be detected by implementing Intrusion detection system (IDS) [44]. Intrusion detection system is the technique of analysing and monitoring network traffic to detect packets flow anomalies. According to [44] intrusion detection can also be implemented using Naïve bayes, Radial basis and rotation forest to discovered intruder network access pattern. Network security threats can be detected by using various types of IDS such as network-based IDS (NIDS), host intrusion detection system(HIDS), network behaviour anomaly detection. These Intrusion Detection Systems apply signature-based detection, anomaly-based detection and stateful protocol inspection to analyse and detect potentially unwanted traffic data and DoS threats. Mukkamala developed an IDS based on Multivariate Adaptive Regression Splines (MARS). It excels at finding optimal variable transformations and interactions, and the complex data structure that often hides in high-dimensional data on the internet. However, new emerging DoS and DDoS attacks require new prediction and detection techniques.

III. Data Collection

The data that were required for the paper include the IP addresses sending request and the bandwidth of the network. The IP addresses were collected and stored in a database by an

IP_Collecting agent. This IP addresses were randomly generated for the purpose of this paper. The bandwidth was pre-entered and an agent was assigned to monitor bandwidth consumption rate against the set threshold. This constituted the network need and formed the basis for the action of the entire detection scheme.

A. System Design

The information gathered by each component of the solution was stored in a relational database table. The relational database was designed using Microsoft Access. The solution consisted of three major parts handled by three agents. The first scheme in the detection mechanism is an agent that stores all Internet protocol (IP) addresses visiting the network. This part receives the addresses visiting a node and updates a record whenever it identifies a new address. The second part of the application is an agent that stores the total bandwidth capacity of the network. The total bandwidth capacity consumed will be monitored per time in relation to some allowable maximum of the total bandwidth. The allowable bandwidth is a fraction of the total bandwidth of the network that is paid for and received from the Internet Service Provider (ISP). The bandwidth threshold is a maximum above which the action agent blocks the network.

B. IP_Collecting Agent

The IP_collecting agent is a specialized agent whose responsibility is to check all IP addresses visiting and/or sending request to the node as shown in Figure 2. It extracts all IP addresses, updates a record when it detects a new address. It will also display a history of all the IP addresses that have previously visited a node. It shows the time as well as the date they visited the network. In addition to monitoring the addresses required and displaying them, the collector agent also contains a method that would be activated in case of a system bridge or perceived attack.

C. Timer Agent

Bandwidth consumption in the network was monitored by this agent as shown in Figure 3. It systematically observes all the in-flow traffic to the network. It monitors the amount of bandwidth consumed in relation to a threshold. The bandwidth threshold is preset as the maximum over which the overall usage should not exceed. This agent was called the threshold agent. It is responsible for checking the overall resource usage of the network. In this case, the resource is the bandwidth. Whenever, the threshold is exceeded, this agent alerts a third agent called the action agent.

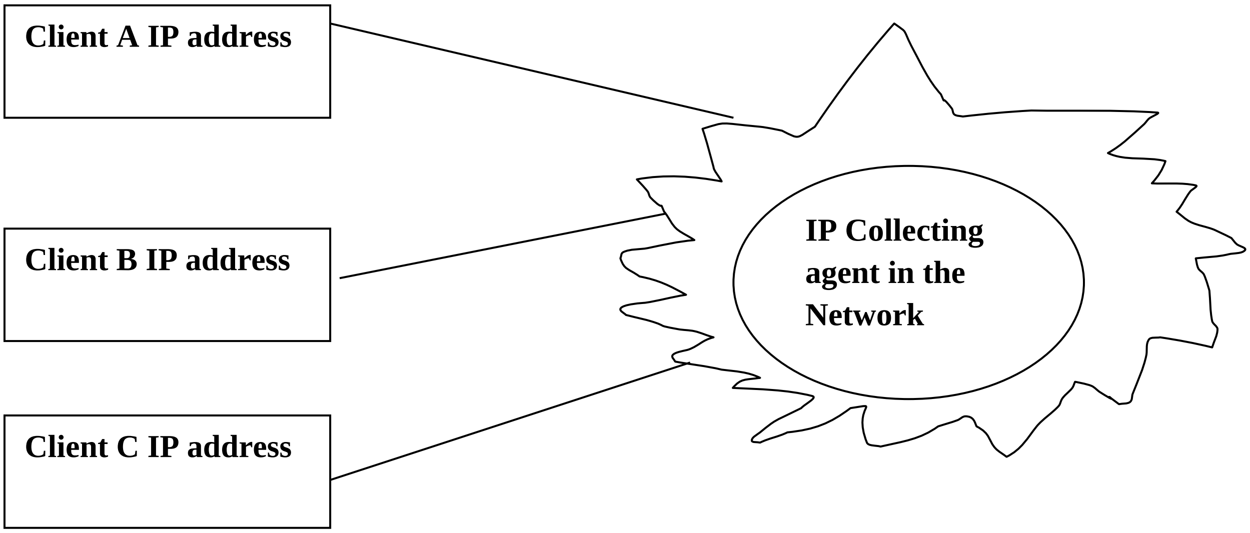


Fig. 2. IP_Collection Agent.

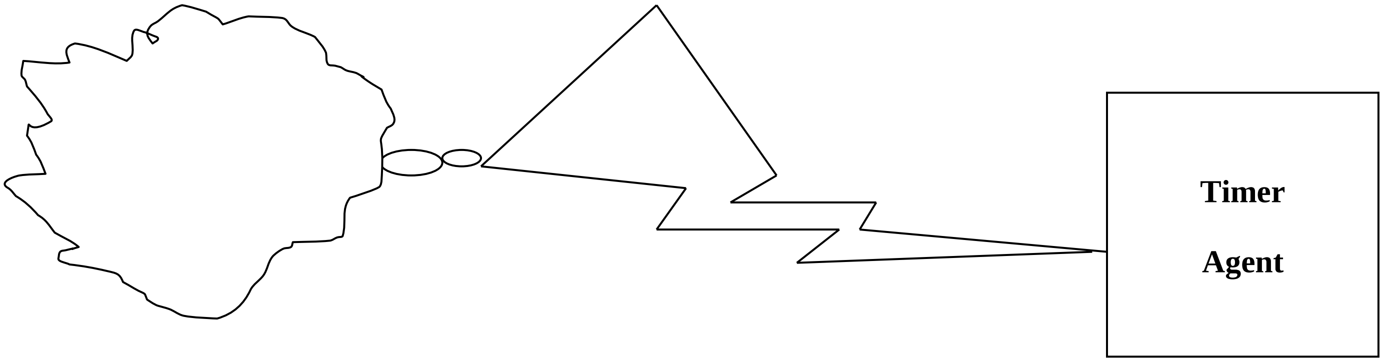


Fig. 3. Timer Agent Listening to the Incoming Traffic.

The threshold agent will broadcast or share this information if and only if the defined threshold is exceeded. This happens whether or not the rate is significantly exceeded or not. A threshold is set in order to reduce the level of false alerts that is, to a reasonable extent. To differentiate between a normal rise in usage and anomaly. The use of threshold also provides a baseline for monitoring this system. It is important to note that the agents' action is based on anomaly. In which case, it is the change in network usage. The agent monitors a particular node and ensures that the bandwidth could not be consumed more than a certain threshold. As soon as bandwidth consumption reaches the stipulated threshold, it alerts the action agent.

D. Action Agent

The third agent in this detection scheme is the action agent. The action agent is usually dormant most of the time until there is a security bridge. It is activated by the threshold agent. When activated, the action agent requests for the IP addresses that visited the network most recently and disconnects all such requests even if they have been granted. The action of the agent and its activities is dependent on the aforementioned agents. A monitoring agent is assigned to each node on the network.

E. System Algorithm

Let X_p be the number of new IP addresses visiting the network at a particular time T. On the other hand, let X_m be the average number of IP addresses that have visited the network over time. Under normal circumstances, the number of new IP addresses should be small for any time interval h. The monitoring agent calculates this value each time a new IP is detected. Whenever X_p is greater than X_m, the Threshold is compared to the bandwidth that has been consumed. If the bandwidth usage is greater than the threshold, then the action agent would be activated. Another variable alert time A_t, is the time the monitoring or threshold agent has to wait to alert the action agent.

$A_t = X_p – X_m$ where A_t is in seconds.

```
Collecting agent listens to IP addresses {
    If {new address is found, update database}
    else {keep monitoring IP addresses}
}
Monitoring agent listens to incoming traffic {
    If ( incoming traffic is greater than bandwidth threshold)
    Then (Performs the calculation algorithm described above and alerts action agent at such time)
    else (continue listening to traffic)
}
Action agent is activated by the monitoring agent {
    Gets the most currently updated IP addresses and disconnects them from the network.
}
```

The Unified Modeling Language diagram shows what part of the system may be activated and what is or should be expected as shown in figure 4.

3.7. UML USE CASE Diagram

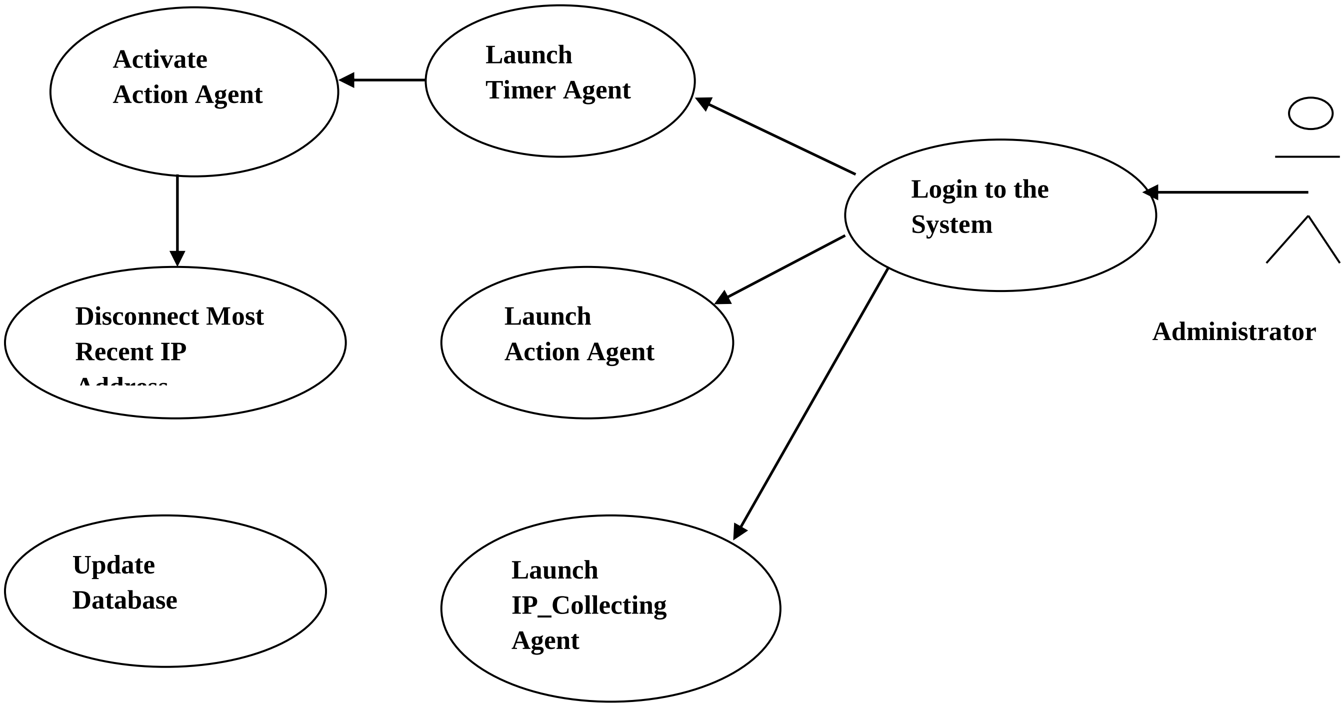


Fig. 4. Use Case Diagram for Accessing the System.

3.9. SYSTEM ARCHITECTURE

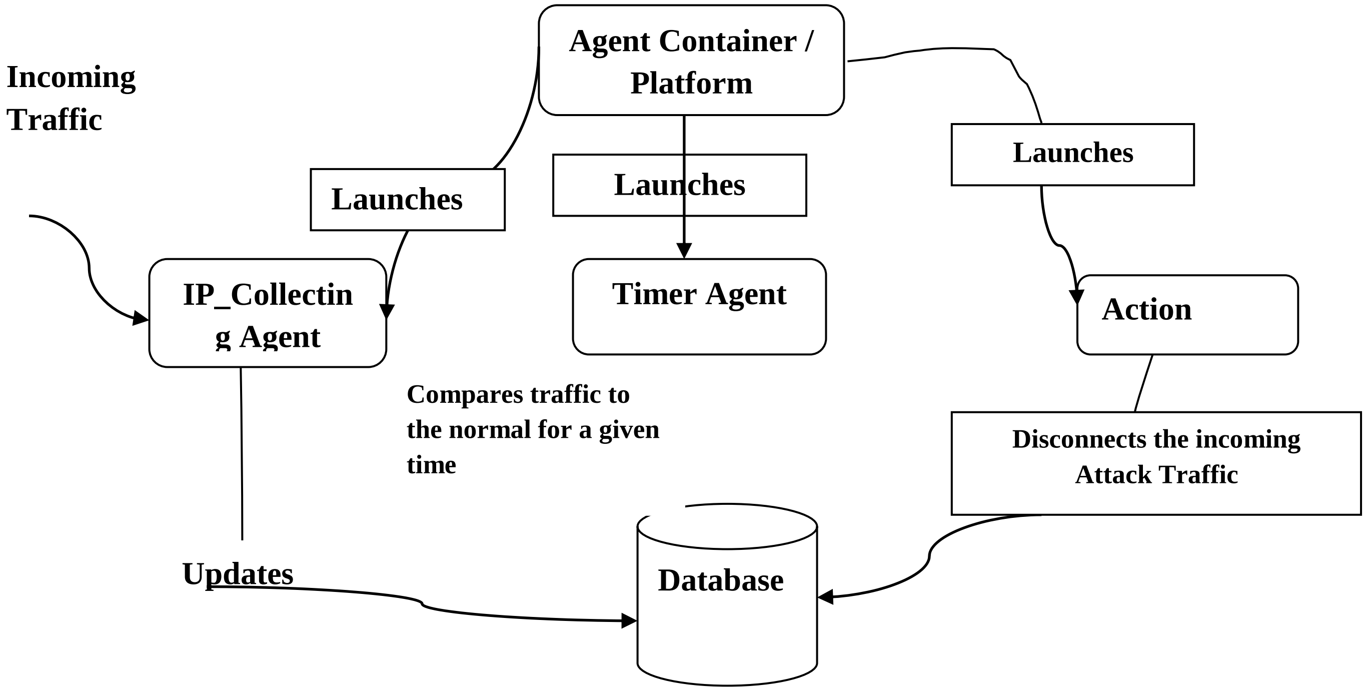

Incoming Traffic
Agent Container / Platform
Launches
Launches
Launches
IP_Collecting Agent
Timer Agent
Action
Compares traffic to the normal for a given time
Disconnects the incoming Attack Traffic
Updates
Database

Fig. 5. System Architecture.

F. Stages in Navigating through the Application

1) Administrative login form where the network administrator logs in with password

2) Start the agent container and launch agents

3) The data monitoring agent is dispatched to monitor incoming request and store fresh IP addresses if not already in the database.

4) The bandwidth monitoring agent is launched to monitor incoming traffic.

5) The agents return their findings.

6) Decision is made based on the reports of the agents.

7) If an attack is suspected, the monitoring agent blocks all new IP addresses making request or disconnects those that have been honored.

The System architecture shows the interaction of various parts of the system and how they interact with one another as shown in figure 5. The agent platform houses all agents naturally. The agents, data collection agent, threshold agent and the action agent were all launched from here. More agents could also be created from the container. This is usually from the JADE platform. Immediately, the agents are launched, the monitoring and threshold agents are active, monitoring the visiting IP addresses and checking bandwidth consumption respectively. The monitoring agent observes the visiting IP addresses and updates the record whenever a new address is encountered. However, the Action agent is inactive for the time until an attack is detected. Whenever the threshold is exceeded, the monitoring agent activates the action agent. When activated, the action agent checks and disconnects all new IP addresses visiting the network for the first time from the updated database.

Normally, an anomaly based detection scheme encounters some challenges. Anomaly based DoS attack detection mechanisms analyse the normal behavior in a system and aim to detect attacks via identifying significant deviation from a normal behavior. Compared to signature based detection approaches, they can discover previously unseen attacks. The challenge therefore is in determining the threshold for anomalous behavior. A model that uses a tight threshold for legitimate behavior in the system may wrongly label normal behavior as malicious (false positive), whereas a loose threshold may lead to many attacks go undetected (false negative). This shortcoming is combated to an extent through the monitoring agent. It is assumed that when an attack is launched, the addresses used to launch the attack are new to network. In order to prevent total breakdown of system, recent requests from new IP addresses would be blocked instead.

G. Network Interface

The user interface is used to launch the network monitor that will display the activity screen. It also contains a button that will display the log details as shown in figure 6.

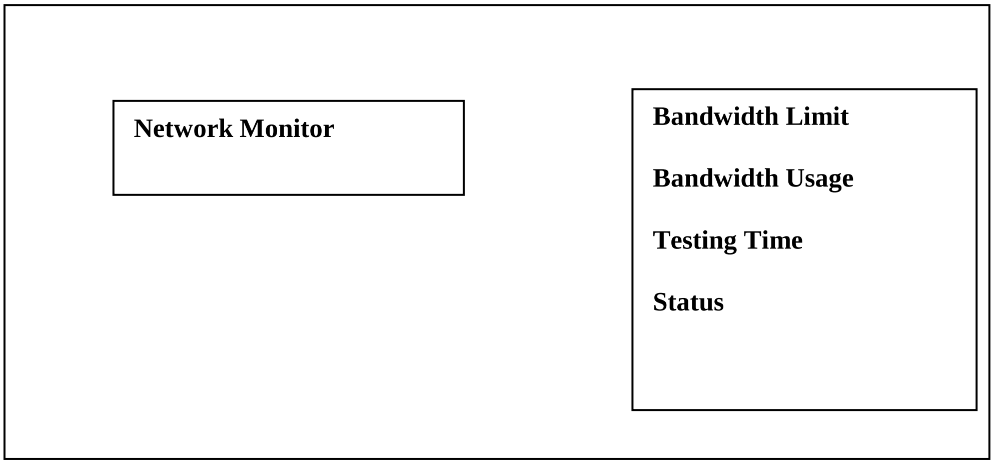


Fig. 6. Network Monitor Interface.

H. Activity Screen

The activity screen shows the IP addresses visiting the network as shown in figure 7. It displays the result of the monitoring agent at any instance. On the activity form, there is a start button to launch the agent. It also contains a start screen to stop an agent. On starting the form, an updated list is received. The activity screen also shows the bandwidth threshold as well as the bandwidth consumed at that time. The information displayed includes:

1) The IP address using or coming into the network.
2) The Time the address visited the network and
3) The date it was visited

I. Log Details

The event log could be viewed through a form interface. It is a report simply to display all the IP addresses as well as the time and date they visited a particular node. It is used by the network administrator to view the history of the IP addresses visiting the network and the particular node. It also stores registered IP addresses which would be used to decide on which IP address.

IV. System Front End

This form provides restricted access to the system. It ensures the system’s integrity and prevents unauthorized access. Providing the correct username and password allows the user to gain access into the system as shown in figure 8.

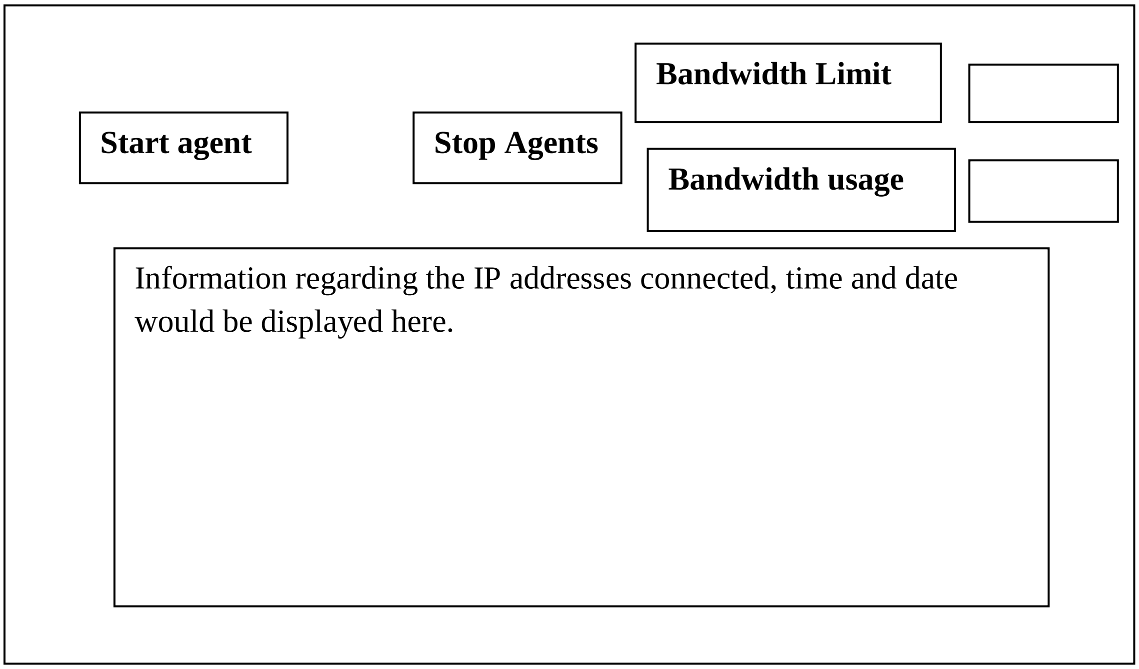
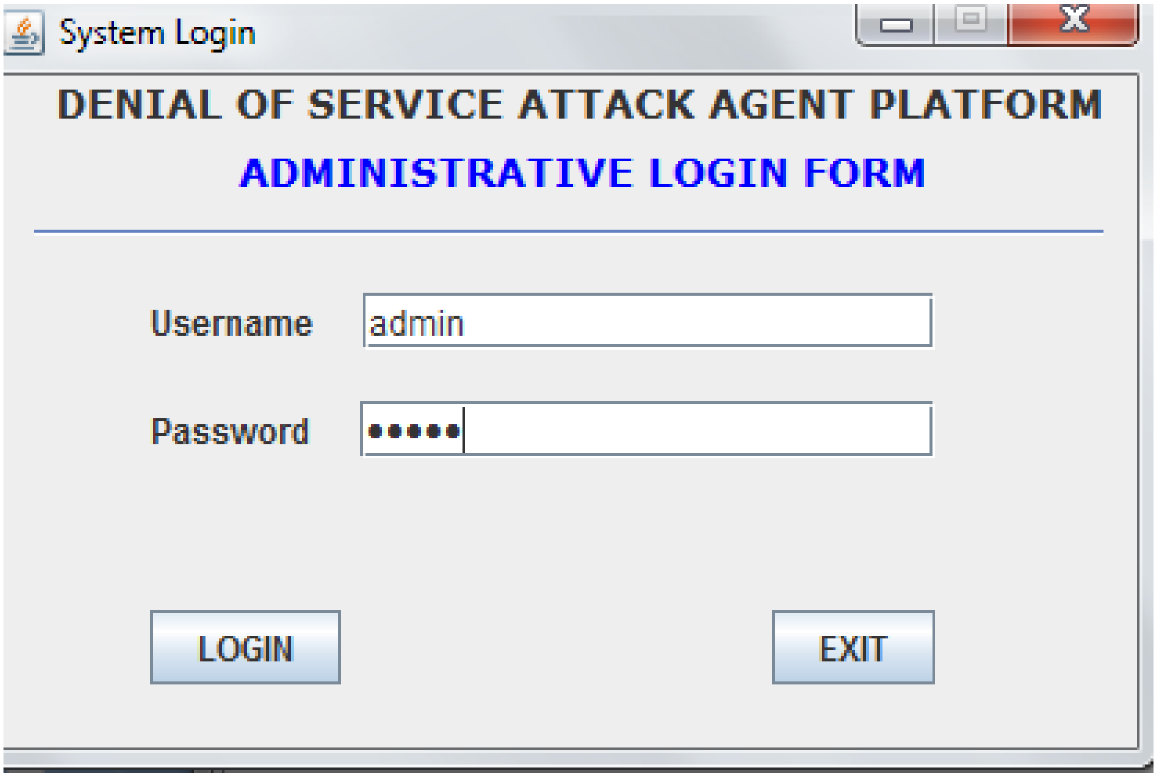

Start agent
Stop Agents
Bandwidth Limit
Bandwidth usage
Information regarding the IP addresses connected, time and date would be displayed here.

Fig. 7. Activity Form.



System Login

DENIAL OF SERVICE ATTACK AGENT PLATFORM

ADMINISTRATIVE LOGIN FORM

Username admin

Password •••••

Fig. 8. Administrative Login Form.

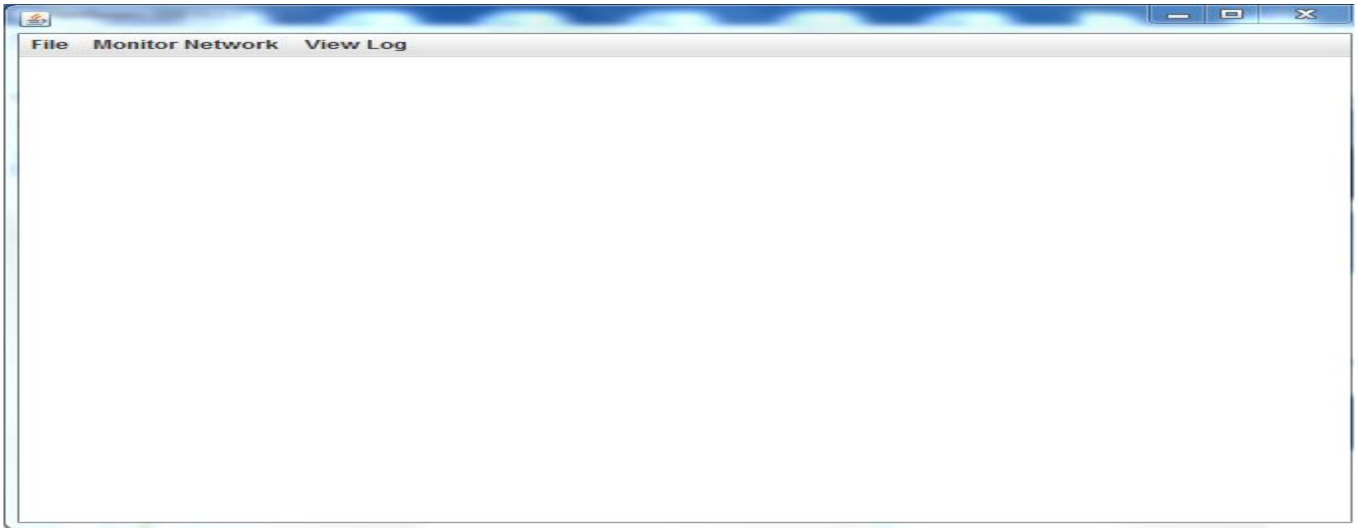


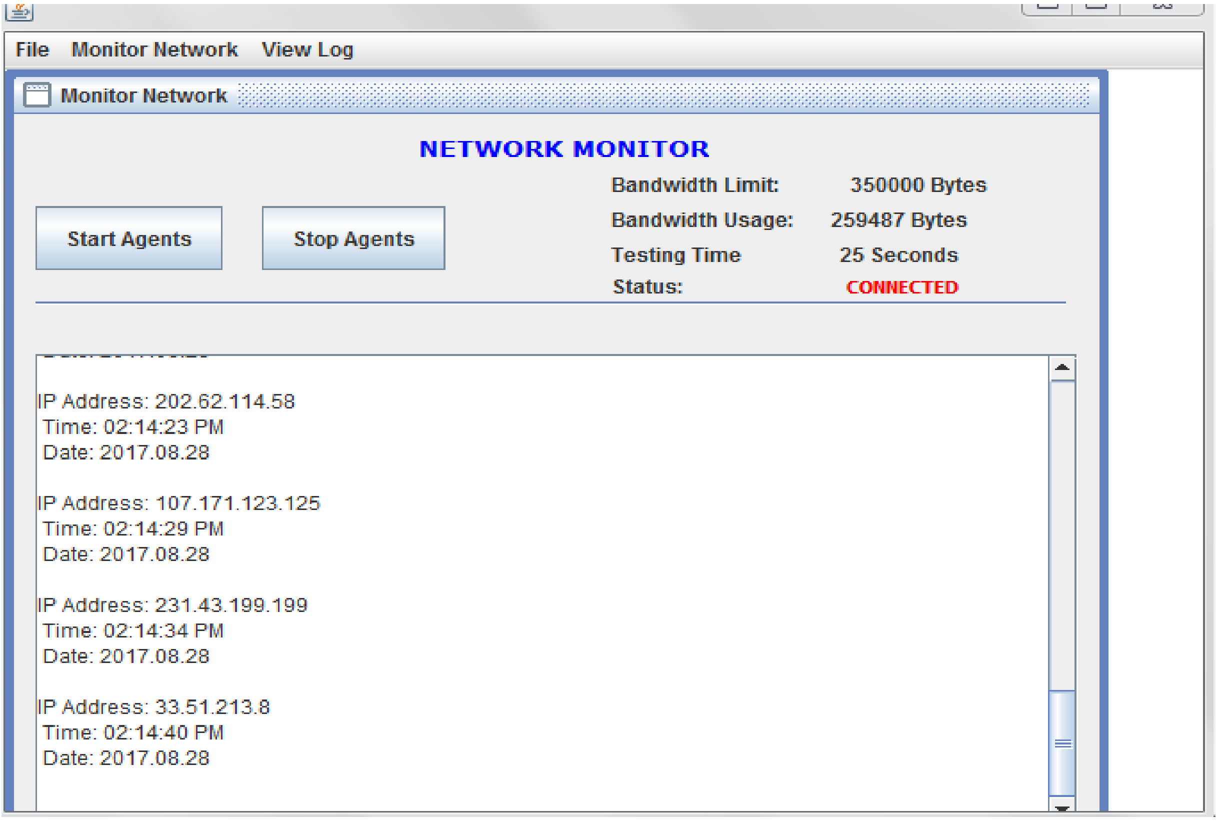
Fig. 9. The Network Monitor before the Agents are Initiated.

A. *System Interface*

In order to launch or stop the system after access is obtained, figure 9 interface allows a user/administrator to either launch the network monitor which is already connected to the JADE environment from the agent platform itself. This interface was designed to provide a level of abstraction for the entire system. However, creating more agents and other maintenance tasks are done from the agent platform.

B. *Network Monitor*

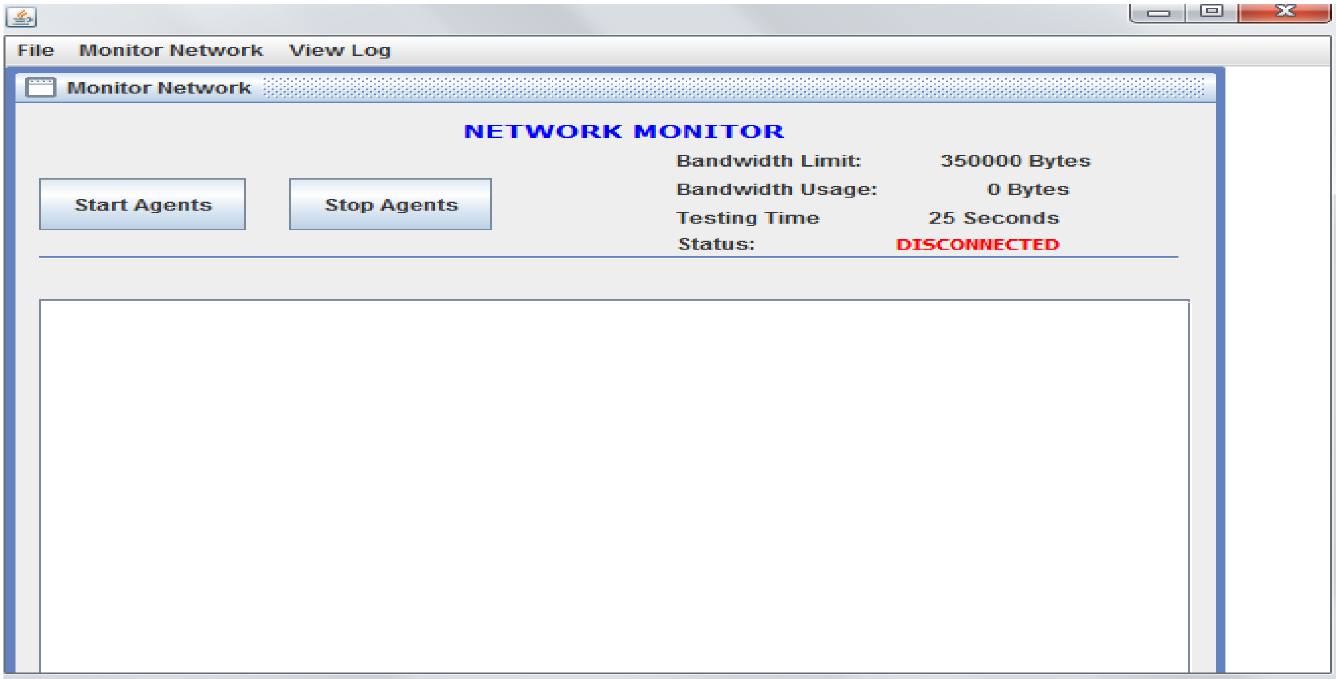
In this section, the IP_Collecting agent is launched. This also means the whole system is started. The “start agent” button initiates the IP_Agent and the Timer Agent as shown in figure 10.



Bandwidth Limit: 350000 Bytes
Bandwidth Usage: 259487 Bytes
Testing Time 25 Seconds
Status: CONNECTED

```
IP Address: 202.62.114.58
 Time: 02:14:23 PM
 Date: 2017.08.28

IP Address: 107.171.123.125
 Time: 02:14:29 PM
 Date: 2017.08.28

IP Address: 231.43.199.199
 Time: 02:14:34 PM
 Date: 2017.08.28

IP Address: 33.51.213.8
 Time: 02:14:40 PM
 Date: 2017.08.28
```

Fig. 10. Network Monitor before the Agents have been Launched.



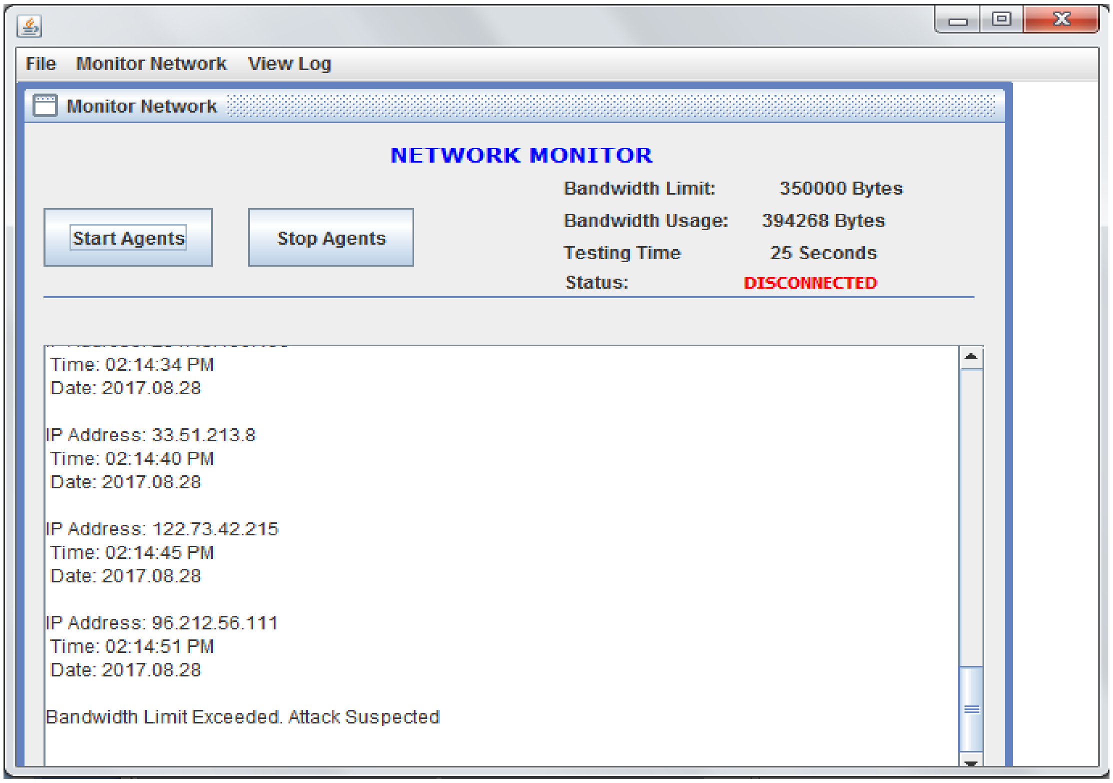
Bandwidth Limit: 350000 Bytes
Bandwidth Usage: 0 Bytes
Testing Time 25 Seconds
Status: DISCONNECTED

Fig. 11. The Network Monitor after the Agents are Initiated.

Figure 11 shows a simulated activity where the bandwidth is being consumed. It represents a normal traffic.

C. Network Monitor Showing a Simulated Denial of Service Attack

Figure 12 shows a faster rate of bandwidth consumption within the observed period of time in which a Dos attack is suspected using the DoS attack detection rules.

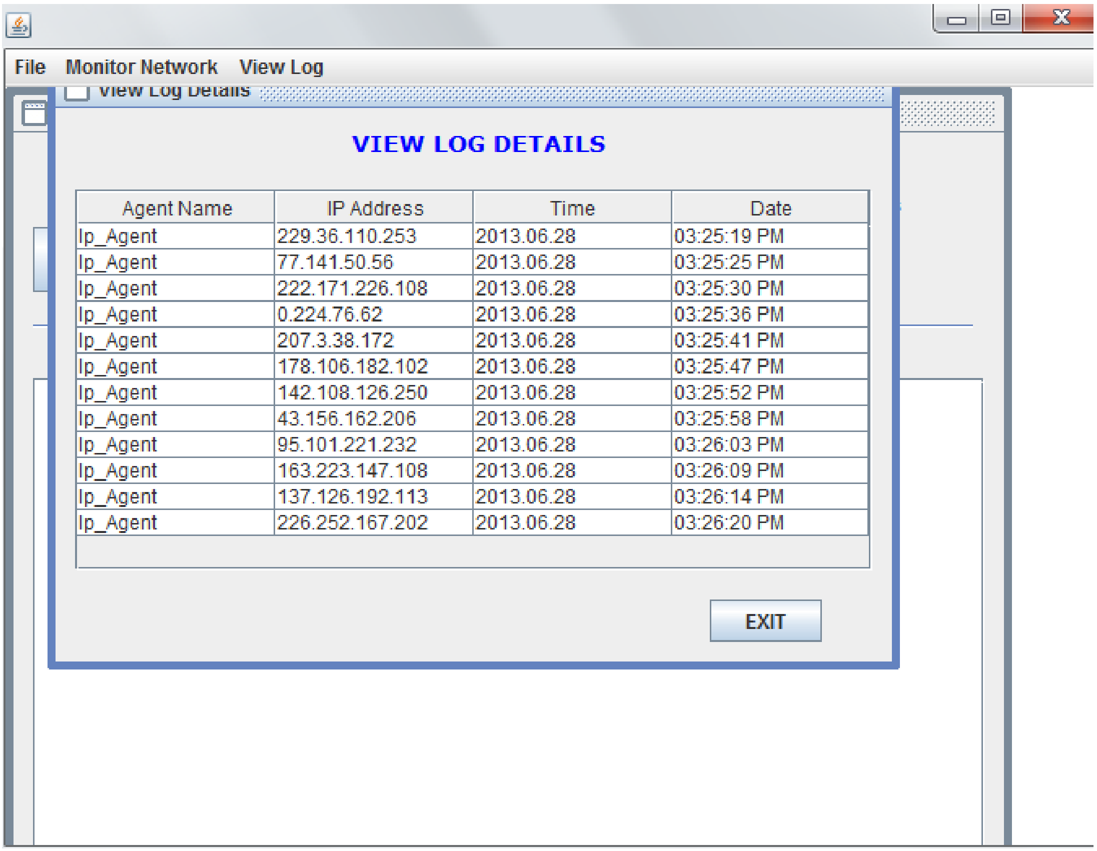


NETWORK MONITOR

Bandwidth Limit: 350000 Bytes
Bandwidth Usage: 394268 Bytes
Testing Time 25 Seconds
Status: DISCONNECTED

```
 Time: 02:14:34 PM
 Date: 2017.08.28

IP Address: 33.51.213.8
 Time: 02:14:40 PM
 Date: 2017.08.28

IP Address: 122.73.42.215
 Time: 02:14:45 PM
 Date: 2017.08.28

IP Address: 96.212.56.111
 Time: 02:14:51 PM
 Date: 2017.08.28

Bandwidth Limit Exceeded. Attack Suspected
```

Fig. 12. Network Monitor Showing an Attack Traffic.



VIEW LOG DETAILS

Agent Name	IP Address	Time	Date
Ip_Agent	229.36.110.253	2013.06.28	03:25:19 PM
Ip_Agent	77.141.50.56	2013.06.28	03:25:25 PM
Ip_Agent	222.171.226.108	2013.06.28	03:25:30 PM
Ip_Agent	0.224.76.62	2013.06.28	03:25:36 PM
Ip_Agent	207.3.38.172	2013.06.28	03:25:41 PM
Ip_Agent	178.106.182.102	2013.06.28	03:25:47 PM
Ip_Agent	142.108.126.250	2013.06.28	03:25:52 PM
Ip_Agent	43.156.162.206	2013.06.28	03:25:58 PM
Ip_Agent	95.101.221.232	2013.06.28	03:26:03 PM
Ip_Agent	163.223.147.108	2013.06.28	03:26:09 PM
Ip_Agent	137.126.192.113	2013.06.28	03:26:14 PM
Ip_Agent	226.252.167.202	2013.06.28	03:26:20 PM

Fig. 13. IP Addresses Record.

D. Log Details

The system also keeps a record of IP addresses that have visited the network which can be accessed from the view log details menu as shown in figure 13

E. Results and Discussions

The result of this paper is a successful simulation of a Denial of service attack where a resource (in this case, a hypothetical bandwidth) was quickly consumed within an observed period of time. The result from the simulated attack traffic was compared against a control (i.e, an expected bandwidth consumption rate) in order to show what could happen in a DoS attack situation. In this paper, the amount of resource consumed is measured as well as the time it is consumed using the DoS detection rules. For the experiment, 350000 bytes was set as a threshold which may vary among organizations depending on the capacity of the network. If available resource is consumed within the test time i.e. 25 seconds then an attack is suspected and all incoming as well as honored requests are terminated. When a resource becomes unavailable, other users who may be legitimate users are prevented from accessing such resource. Scenarios like this can disrupt the proper functioning of the system and in some cases pave the way for more serious forms of attacks.

V. CONCLUSION

Detection of DoS attack was demonstrated in this paper using mobile agent technology with focus on bandwidth consumption using hypothetical traffic. If the amount of resource consumed can be monitored and controlled by keeping track of the traffic in a network, then the devastation and loss caused by DoS attacks can be highly reduced if not totally eradicated. Hence, providing a threshold for the amount of resource that should be consumed would prevent computer networks from crashing totally before an action can be taken.

A drawback for this study is the temporal disconnection of requests for those that have been granted and the incoming ones. This may be solved by providing a better response method. A record can be maintained for visiting IP addresses without depending on internet service providers. The records obtained can now be further analysed to see the patterns of suspected IP packets and other methods for response can now be formulated by monitoring a particular network.

REFERENCES

[1] Kessler, Gary C. 2000. Defenses Against Distributed Denial of Service Attacks. http://www.garykessler.net/library/ddos.html

[2] Gibson and Steve. 2001. The Strange Tale of the Denial of Service Attacks Against GRC.COM. http://www.crime-research.org/library/grcdos.pdf. Gibson Research Corporation.

[3] Andersen, D. and Balakrishnan, H. and Kaashoek, F. and Morris, R. 2001. Resilient Overlay Networks. ACM.

[4] Peng, T and Leckie, C and Ramamohanarao, K. (2007). Survey of Network-Based Defense Mechanisms Counteringthe DoS and DDoS Problems. http://doi.acm.org/10.1145/1216370.1216373. ACMJ258-03.

[5] Mukherjee B, Todd L, Levitt K(1994). “Network intrusion detection, IEEE Network, Vol. 8, No. 3, pp.26–41, 1994

[6] Teng H. S. Teng, K. Chen and S. C. Lu(1990). ”Security Audit Trail Analysis Using Inductively Generated Predictive Rules”. In Proceedings of the 11th National Conference on Artificial Intelligence Applications, pages 24-29, IEEE, IEEE Service Center, Piscataway, NJ, March 1990.

[7] Porras, 1992 ”STAT: A State Transition Analysis Tool for Intrusion Detection”. Master’s Thesis, Computer Science Dept., University of California, Santa Barbara, July 1992.

[8] Dunigan T and Hinkel G (1999), “Intrusion detection and intrusion prevention on a large network: A case study”, Proc. of workshop on intrusion detection and network monitoring, 1999.

[9] Ilgun, 1993, "USTAT: A Real-time Intrusion Detection System for UNIX," 16-28. Proceedings of the 1993 Computer Society Symposium on Research in Security and Privacy. Oakland, California, May 24-26, 1993. Los Alamitos, CA: IEEE Computer Society Press, 1993.

[10] Garvey T. D. and Lunt T. F (1991), ”Model based intrusion detection”. In Proceedings of the 14th National Computer Security Conference, pages 372-385, October 1991.

[11] S. Kumar and E. H. Spafford (1994). ”An Application of Pattern Matching in Intrusion Detection”. Technical Report CSD-TR-94-013, Purdue University, 1994.

[12] S. Kumar (1995). ”Classification and Detection of Computer Intrusions”. PhD Thesis, Department of Computer Science, Purdue University, August 1995.

[13] Mukkamala S,. Sung A.H and Abraham A. (2003), ”Intrusion Detection Using Ensemble of Soft Computing Paradigms”, Third International Conference on Intelligent Systems Design and Applications, Intelligent Systems Design and Applications, Advances in Soft Computing, Springer Verlag, Germany, pp. 239-248, 2003.

[14] Shah K., Dave N., Chavan S., Mukherjee S. (2004), A. Abraham and S. Sanyal, ”Adaptive Neuro-Fuzzy Intrusion Detection System”, IEEE International Conference on Information Technology: Coding and Computing (ITCC'04), USA, IEEE Computer Society, Volume 1, pp. 70-74, 2004.

[15] S Staniford-Chen, S, B Tung, and D Schnackenberg(1998), “The Common Intrusion Detection Framework (CIDF)”. Proc. Information Survivability Workshop, Orlando FL, October 1998.

[16] Ning P, Jajodia S, Wang X(2002), “Design and implementation of a decentralized prototype system for detecting distributed attacks”, Computer Communications, Vol. 25, pp. 1374-1391, 2002.

[17] Kim, Y. Cheong L., Chuah C. And Jonathan H. (2004) “Packetscore: Statistical-Based Overload Control Against Distributed Denial-Of-Service Attacks” IEEE INFOCOM 2004 ,The 23rd Annual Joint Conference of the IEEE Computer and Communications Societies, Hong Kong, China, March 7-11, 2004. IEEE.

[18] Dittrich, D. Dietrich, S. and Long, N.(2000). Analyzing distributed denial of service attack tools: The shaft case. In Proceedings of 14th Systems Administration Conference. New Orleans, Louisiana, USA, 329-339.

[19] Dobbins R. and Morales C.(2010), “ Worldwide lnfrastructure Security Report, “Arbor Networks, Vol VII, Massachusetts, USA.

[20] Gavrilis, D. and Dermatas, E. (2005) Real-time detection of distributed denial-of-service attacks using RBF networks and statistical features. Computer Networks and ISDN Systems, 48, 235–245.

[21] Kampanakis P., Perros H., and Beyene T. (2014), Sdn-based solutions for moving target defense network protection. In WoWMoM, pages 1–6. IEEE Computer Society.

[22] Debroy S., Calyam P., Nguyen M., Stage A., and Georgiev V. (2016). Frequency-minimal moving target defense using software-defined networking. 2016 International Conference on Computing, Networking and Communications (ICNC), 00:1–6.

[23] Roshan L., Travis N., Nishant C (2018), Dolus: Cyber Defense using Pretense against DDoS Attacks in Cloud Platforms, 19th International Conference on Distributed Computing and Networking, January 4–7, 2018, Varanasi, India. ACM, New York, NY, USA,

[24] Abhilasha, P., & Santosh, W. (2017). Denial-Of-Service Attack Detection Using Artificial Neural Network Based On Genetic Algorithm and Multivariate Correlaton Analysis . *International Journal of Innovative Research in Science, Engineering and Technology*.

[25] Adrien, B., & Martine, B. (2017). A Survey of Denial-of-Service and Distributed Denial of Service Attacks and Defenses in Cloud Computing. *Future Internet*, 1-19.

[26] Alan, S., Richard, E., & Tomasz, R. (2015). Detection of known and unknown DDoS attacks using Artificial Neural Networks. *Science Direct*.

[27] Aleksandar, M. (2016, May 5). *Intrusion Detection Systems*. Retrieved from University of Würzburg: http://www.uni-wuerzburg.de/fileadmin/10030200/IDS.pdf

[28] Charalampos, P., Michalis, M., & Olga, Z. (2017, May 5). Distributed Denial of Service Attacks. *The Internet Protocol Journal*. Retrieved from CISCO: https://www.cisco.com/c/en/us/about/press/internet-protocol-journal/back-issues/table-contents-30/dos-attacks.html

[29] Elliot, M., Ralph, M., Nation, C., & Antony, C. (2017). Fraud Detection in E-Transactions using Deep Neural Networks-A Case of Financial Institutions in Zimbabwe. *International Journal of Science and Research* , 1036-1041.

[30] Felix, L., Stuart, R., Michael, S., & Ljiljana, T. (2000). Distributed Denial of Service Attacks. *IEEE Xplore*.

[31] Jelena, M., & Peter, R. (2002). A Taxonomy of DDoS Attack and DDoS Defense Mechanisms. *citeseerx*.

[32] Jelena, M., & Peter, R. (2014). A taxonomy of DDoS attack and DDoS defense mechanisms. *ACM SIGCOMM Computer Communications Review*, 39-53.

[33] Jun, W., Wang, X., & Lee, B. Y. (2010). Detecting DDoS Attack towards DNS Server Using a Neural Network Classifier. *Springer*, 118-123.

[34] Keyur, C., & Vivek, P. (2015). Distributed Denial of Service(DDoS) Attack Techniques and Prevention on Cloud Environment. *International Journal of Innovations & Advancement in Computer Science*, 1-6.

[35] Kihong, P., & Heejo, L. (2001). On the Effectiveness of Route-Based Packet Filtering for Distributed DoS Attack Prevention in Power-Law Internets. *citeseerx*.

[36] Munivara, P., Rama, M. ,., & Venugopal, R. (2014). DoS and DDoS Attacks: Defense, Detection and Traceback Mechanisms - A Survey. *Global Journal of Computer Science and Technology: ENetwork, Web & Security* .

[37] National CyberSecurity Center. (2018, May 05). *Understanding denial of service (DoS) attacks*. Retrieved from National CyberSecurity Center: https://www.ncsc.gov.uk/guidance/understanding-denial-service-dos-attacks

[38] Nicholas, W. (2002). Potential Strategies for High Speed Active Worms:A Worst Case Analysis. *citeseerx*.

[39] Ogunleye, Adewale, & Alese. (2013). DETECTING DDOS ATTACK IN AN AGENT-BASED VIRTUAL KNOWLEDGE COMMUNITY. *Seria Informatică*.

[40] Oyebode E.O., Fashoto S.G.,Ojesanmi O.A. and Makinde O.E.(2011). Intrusion Detection System for Computer Network Security. Australian Journal of Basic and Applied Sciences, 5(12): 1317-1320, ISSN 1991-8178,2011.

[41] Parneet, K., Manish, K., & Abhinav, B. (2017). A review of detection approaches for distributed denial of service attacks. *Systems Science & Control Engineering An Open Access Journal*, 301-320.

[42] Qijun, G., & Peng, L. (2018, May 05). *Denial of Service Attacks* . Retrieved from Pennsylvania State University : https://s2.ist.psu.edu/ist451/DDoS-Chap-Gu-June-07.pdf

[43] Santos, K., Chandra, S., Phani, R., Dawood, B., & Sudhakar. (2013). Intrusion Detection System- Types and Prevention. *International Journal of Computer Science and Information Technologies*, 77 - 82.

[44] Vangie, B. (2018, May 5). *DDoS attack - Distributed Denial of Service*. Retrieved from Webopedia: https://www.webopedia.com/TERM/D/DDoS_attack.html