BOTNET DETECTION USING FEED FORWARD NETWORK

Seema Shirke[1] , Anish Dhawade[2] , Sophiya Mandhare[3], Sai Surve[4] , Nikunj Sardesai[5]

[1] *Assitant Professor, Information Technology, RMD Sinhgad School of Engineering, Maharashtra, India*
[2]*Student, Information Technology, RMD Sinhgad School of Engineering , Maharashtra, India*
[3]*Student, Information Technology, RMD Sinhgad School of Engineering, Maharashtra, India*
[4]*Student, Information Technology, RMD Sinhgad School of Engineering, Maharashtra, India*
[5]*Student, Information Technology, RMD Sinhgad School of Engineering, Maharashtra, India*

ABSTRACT

Traditional anomaly detection methodologies are not able to efficient deal with the detection of botnet hence a new botnet detection approach is proposed that can uniquely identify botnet activity. A concerned fight against botnets is needed in order to avoid them from becoming a serious threat to global security in forthcoming years. Detects botnet activity based on traffic behavior analysis by classifying network traffic behavior. In this work we aim to fulfill the approach like Dumping the TCP packets, filtering TCP packets, detecting the bots using Artificial Neural Network. It has a strong defense system which can be established in a low cost, provides high security and requires less time to analyze the bots.

Keyword: - *Botnet, Botnet Detection, Feed Forward Network.*

1. INTRODUCTION

A botnet is a collection of computers connected to the Internet which have been compromised and are being controlled remotely by an intruder (the botmaster) via malicious software called bots. While a significant amount of research has been accomplished on botnet analysis and detection, several challenges remain unaddressed, such as the ability to design detectors which can cope with new forms of botnets. Detect botnet activity based on traffic behavior analysis by classifying network traffic behavior using machine learning. Traffic behavior analysis methods do not depend on the packets payload, which means that they can work with encrypted network communication protocols. Network traffic information can usually be easily retrieved from various network devices without affecting significantly network performance or service availability.

One or more methods are used to spread bots to infect large number of hosts. New victims are joining the botnets and as a result rapid spreading of bots takes place.In this process commands are distributed by attackers to victims via bots. There are different control and command protocols and hence the bots are separated into three different categories botnet based on IRC protocol (IRC-based botnet), botnet based on HTTP protocol (HTTP-based botnet) and botnet based on P2P protocol (P2P-based botnet) as in [6].Earlier a majority of proposed systems were specifically designed for IRC botnets but now the focus is mainly on P2P-based and HTTP-based botnets. Machine learning techniques are used to generate botnet detection models which is automatic.[3] One of the most early method used for detection of botnets as honeypots. But this method as not effective as expected. Researchers could not detect bot infection for every occasion. Honeypot ere used to track botnets in the netork for generating an early report for understanding the consequences.[7]In this paper we aim to detect HTTP based botnet activity based on botnet behavior analysis via machine learning approach to detect bots and avoid any malicious activities to be carried out by these bots.

The rest of the structure is as follows:Section 2 summaries the system architecture of botnet detection and the working with the results has been mentioned. Finally, conclusions are drawn at the end.

2.SYSTEM ARCHITECTURE

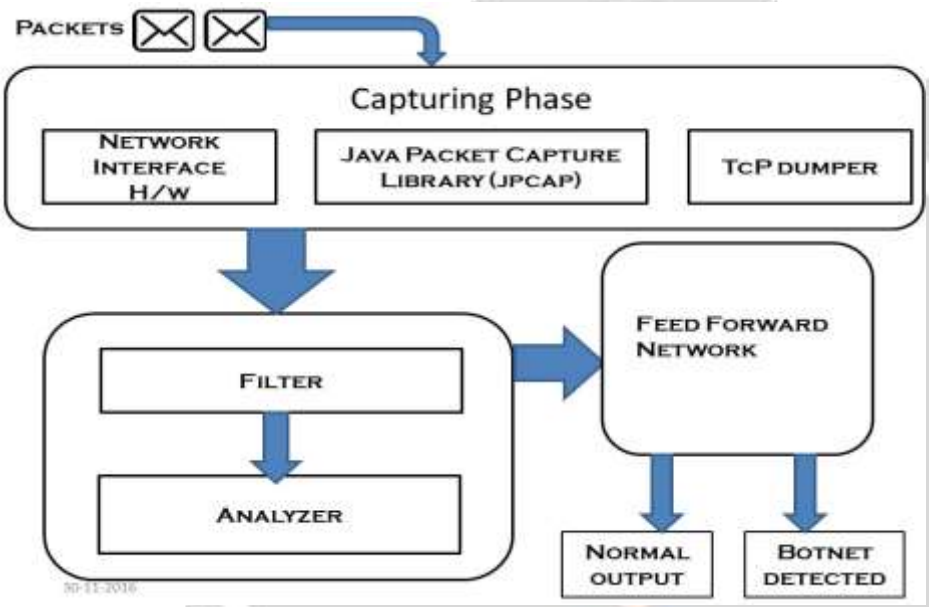
This section includes four main phases for detection of botnets.

1.Capturing Phase

2.Filtering and Analyzing Phase

3.Feed Forward Network

4. Detecting Bots



Capturing Phase:-In this phase, the incoming packets are captured. These packets are captured with the help of JP Cap Library. The incoming packets are initially captured by the Win cap(Windows Packet Capture). These captured packets are then transferred to the JP Cap Library. These packets are then stored with the help of TCP Dumper.

Creating TCP Packet Dumper.

- Using JpCap library dumps the all TCP requests (packet) into the dump file. Dump file size is maximum 2048KB.
- Dump file created on the basis following attribute :

packetIndex,Timeval,sourceAddress,sourceHardwareAddress,sourcePort,destinationAddress,destinationHardwareAddress,destinationPort,sequenceNumber,acknowledgementNumber,flagsPresent,packetPriority,packetLength,offset, TimeToLive.

1.Filtering and Analyzing Phase:-The dump file is created which contains all the information about the incoming TCP packets. It contains source address, destination address, hardware source address, hardware destination address, Average Length of TCP packet, etc.

These dumped files are analyzed on the basis of their Hardware Address, Count of Packet from the Different IP, Average Length of TCP packet, Average difference in consecutive requests (Δ time), Request Ratio, Number of self packets/total packets.

2.Apply TCP Packet Filtering:-We can apply the TCP packet filter on the dump files only for the incoming TCP request.

3.Web Application Firewall.

a) TCP Packet Analyzer

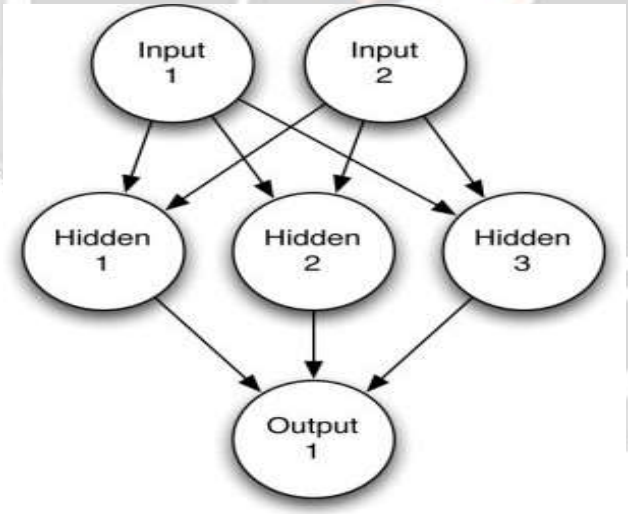
It performs the analysis on dump files and give the following attribute to Feed Forward Network..

- Hardware Address
- Count of Packet from the Different IP
- Average Length of TCP packet
- Average difference in consecutive requests (Δ time).
- Request Ratio
  Number of self packets/total packets

b) Apply Feed Forward Network Algorithm

TCP Packet analyzer output use as the Feed Forward Network Algorithm input the dataset is train on the TCP Packet Analyzer attribute.

Feed Forward Network:-



Stepwise Algorithm Implementation

1. Considering the attributes of the incoming packets these packets are dumped in a file.
2. These dumped files are then analyzed depending on its attributes.
3. Derived attributes are extracted from the dumped file.
4. The derived attributes are provided as an input to the Feed Forward Network.
5. The output of the Feed Forward Network is compared with dataset.
6. Depending on the compared result the Feed Forward Network decides whether the packets is malicious or not, or if it is an attack.
7. If the malicious packets are detected then it is an attack.
8. If examined as an attack the IP address is blocked.

9. If no malicious packets detected, the packets received are normal.

Input Layer:-It is the layer where the packets are received and given as an input to the next layer.

Hidden Layer:- Hidden layer has weights assigned to each of its nodes. The calculations are done using weights of the input layer and the hidden layer and each of its node is assigned with a new weight according to previous calculations.

Output Layer:- The result of hidden layer is given as an input to the output layer. The output obtained from different hidden layer nodes is then compared and the final decision is made.

Detecting Bot:- On the basis of Feed Forward algorithm result make the decision to allow the packets from the particular machine or not which can be Bot.

Algorithms:

Feed Forward Network Algorithm.

Step 1:Then-dimensional weight vectorsw1,w2, . . . ,wm of the m computing units are selected at random. An initial radius r, a learning constant η, and a neighborhood function φ are selected.

Step 2: Select an input vector ξ using the desired probability distribution over the input space.

Step 3: The unit k with the maximum excitation is selected (that is, for which the distance between w_i and ξ is minimal, i = 1, . . .,m).

Step 4: The weight vectors are updated using the neighborhood function and the update rule.

$$\mathbf{w}_i \leftarrow \mathbf{w}_i + \eta\phi(i,k)(\xi - \mathbf{w}_i), \quad \text{for } i = 1, \ldots, m.$$

Step 5: Stop if the maximum number of iterations has been reached; otherwise modify η and φ as scheduled and continue with step 1.

Results

1.Home Page

Image 10.1

Home page is the initial page that gets displayed when the proposed system is executed. It also contains a direct link to the network trafficviewer i.e. the page where the information of the packets is displayed in a tabular format with all its necessary attributes. It also contains a direct link to the statistics.

2.Statistics

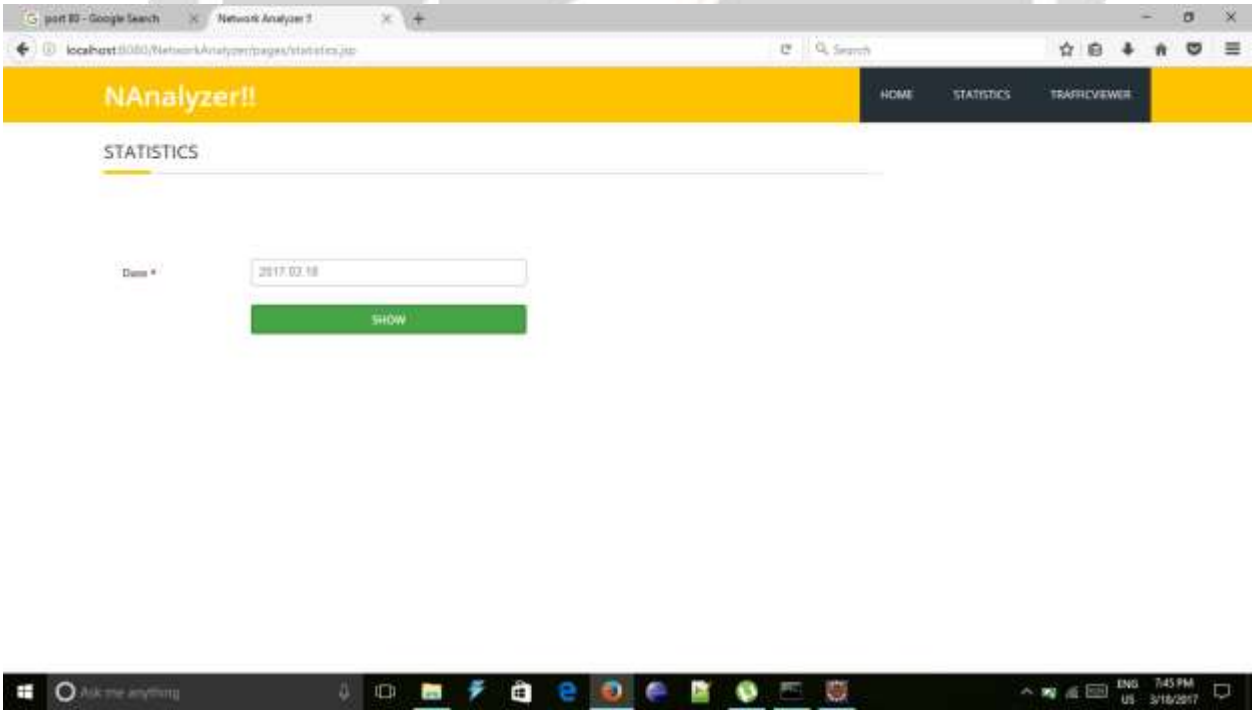


Image 10.2

Initially the home page is displayed and there is link to the statistics. Show button will display the files which are dumped by the TCP Dumper . It will show a file with its file name and the date

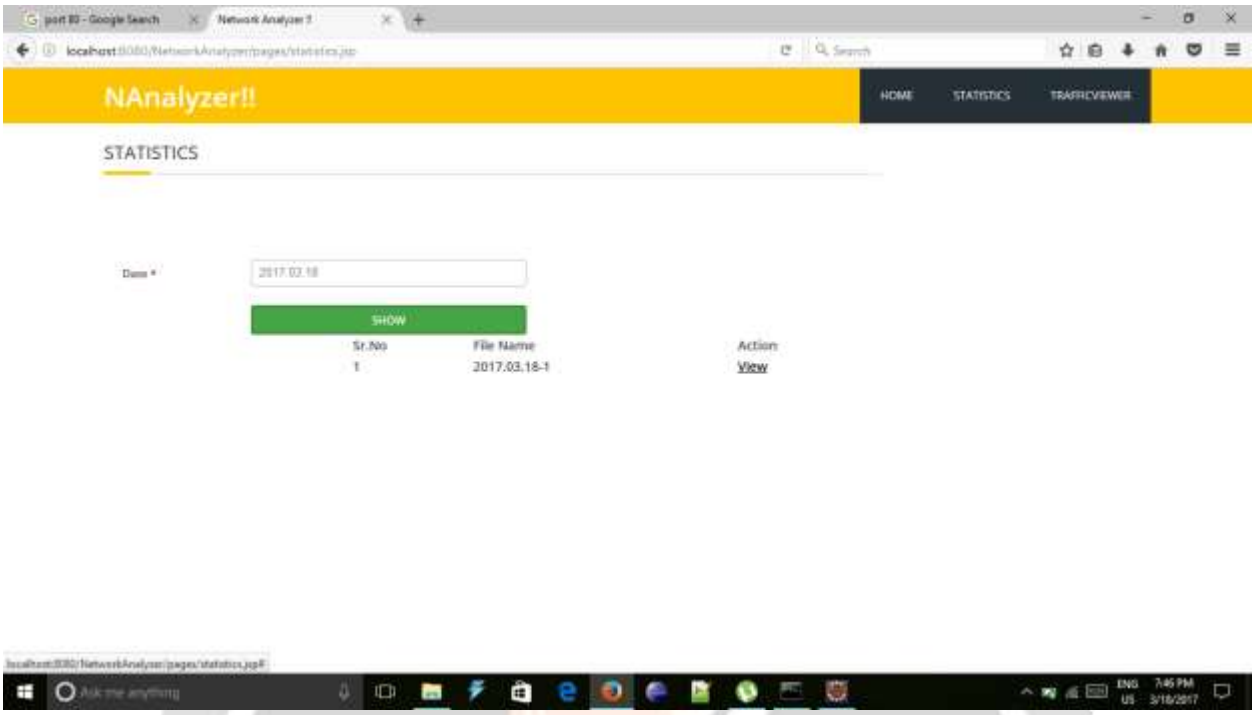


Image 10.3

From the home page the statistics can be viewed. Proposed System will show the present date and a show button below it. The show button will show a file name and its date and a view button. Click the view button and it shows us the packets analyzed on the basis of its attributes as per the date and time.

3.Traffic Viewer

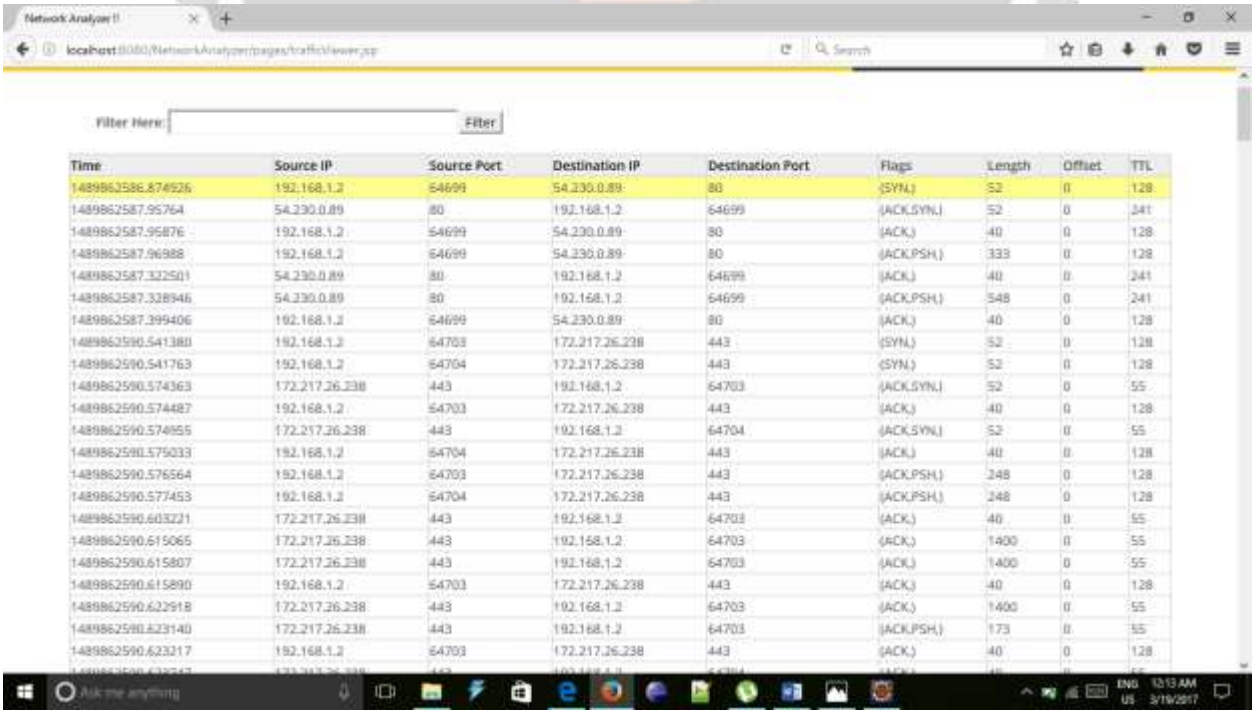


Image 10.4

Traffic Viewer is used to display the core attributes of the dumped packets. It gives the brief description of packets arriving at IP Address of the proposed system with the help of its attributes like time, source IP, source port, destination IP, destination Port, Flags, Length etc.

4.Proposed System Current IP Address

This is the IP address of the proposed system in the current network. It means that only the packets receiving at this IP Address will be dumped and not the outgoing packets from this IP Address.

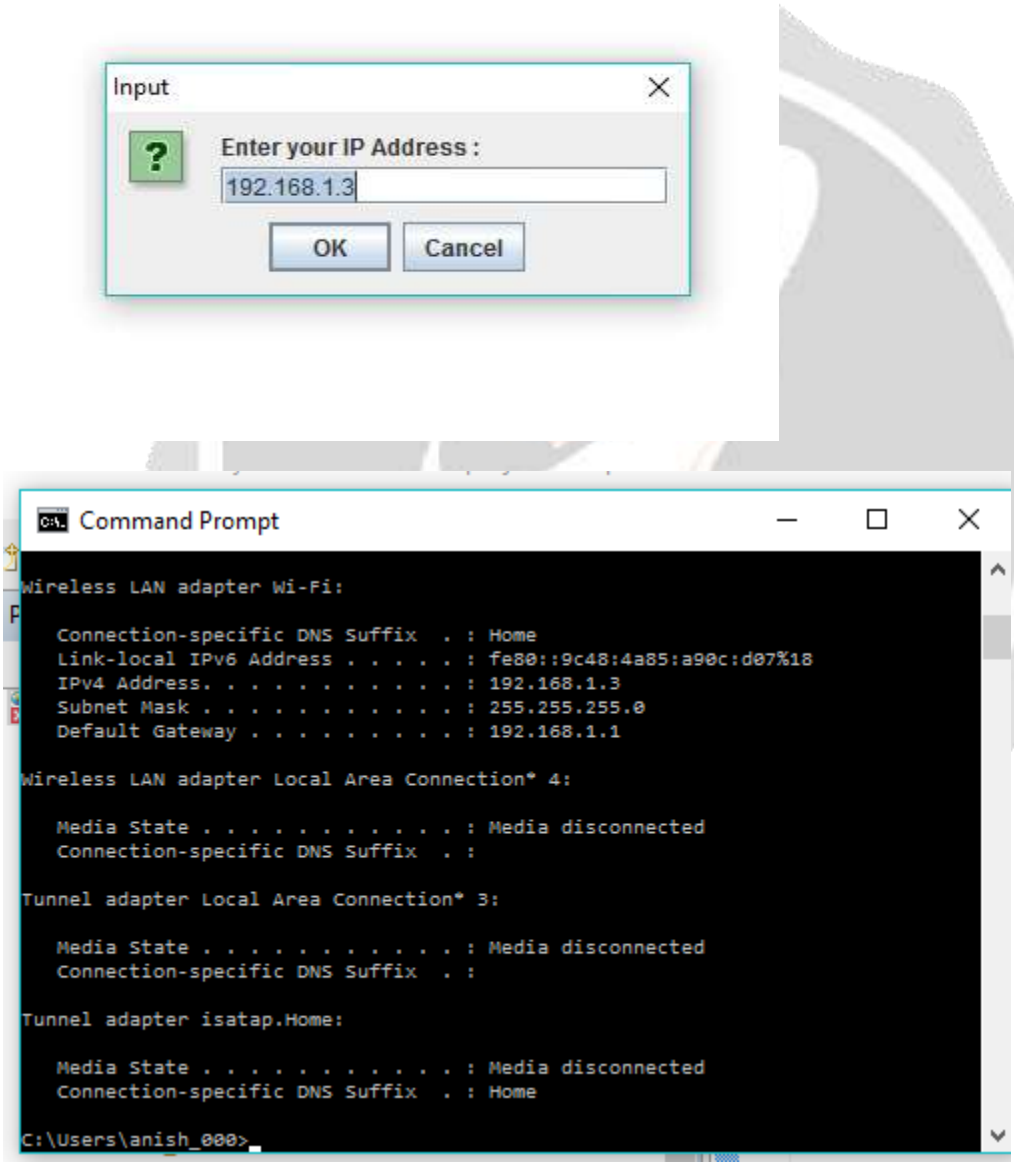


Image 10.5

IP Address shown above by using the command prompt. Command→ ipconfig will display the IP Address. It will show the IP Address of the machine and its Gateway as well. We should compare The IP Address of the command prompt with the proposed system. If it matches then the system is working correctly.

5.Normal Output

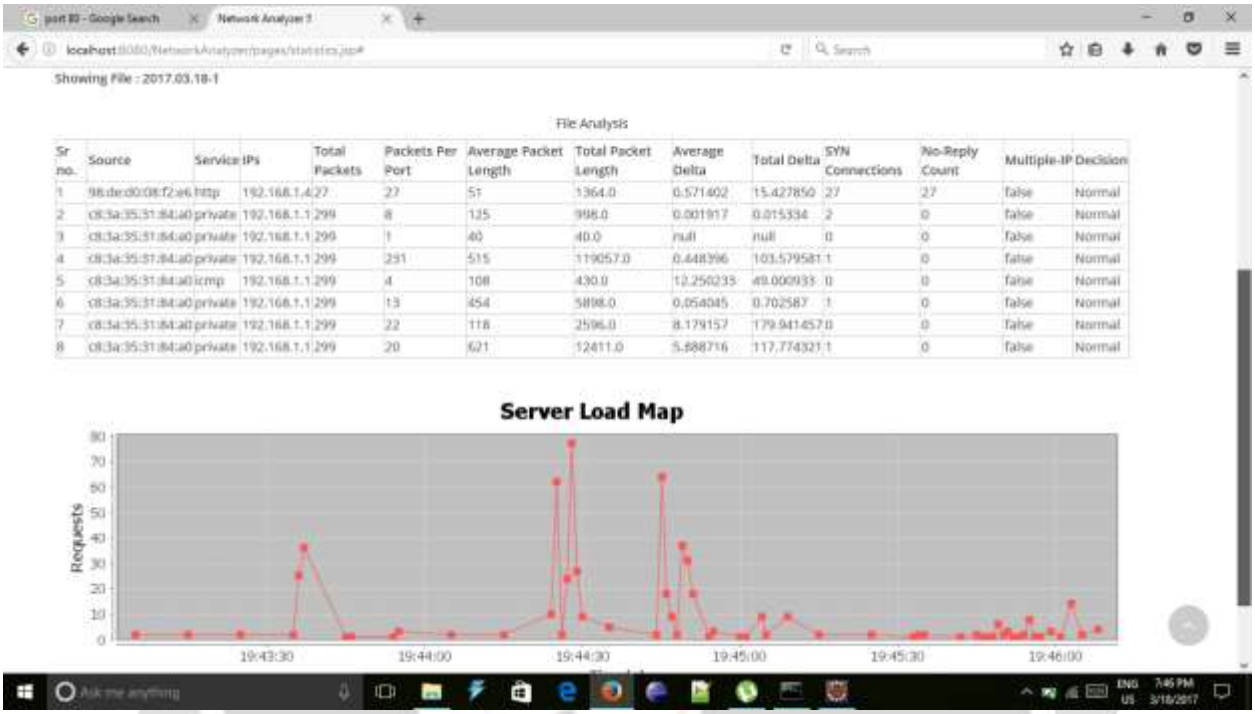


Image 10.6

It shows that all the packets received and analyzed by the proposed system are normal and there is no attack up till now. It also show the server load with the help of graphical representation. No of requests and the receiving time are the Y and X axis in the graph respectively. It is displayed that at a specific time it is very high. It means that the packets received at a very high speed. Depending upon the speed the graph will vary.

6.Attacker's IP Address

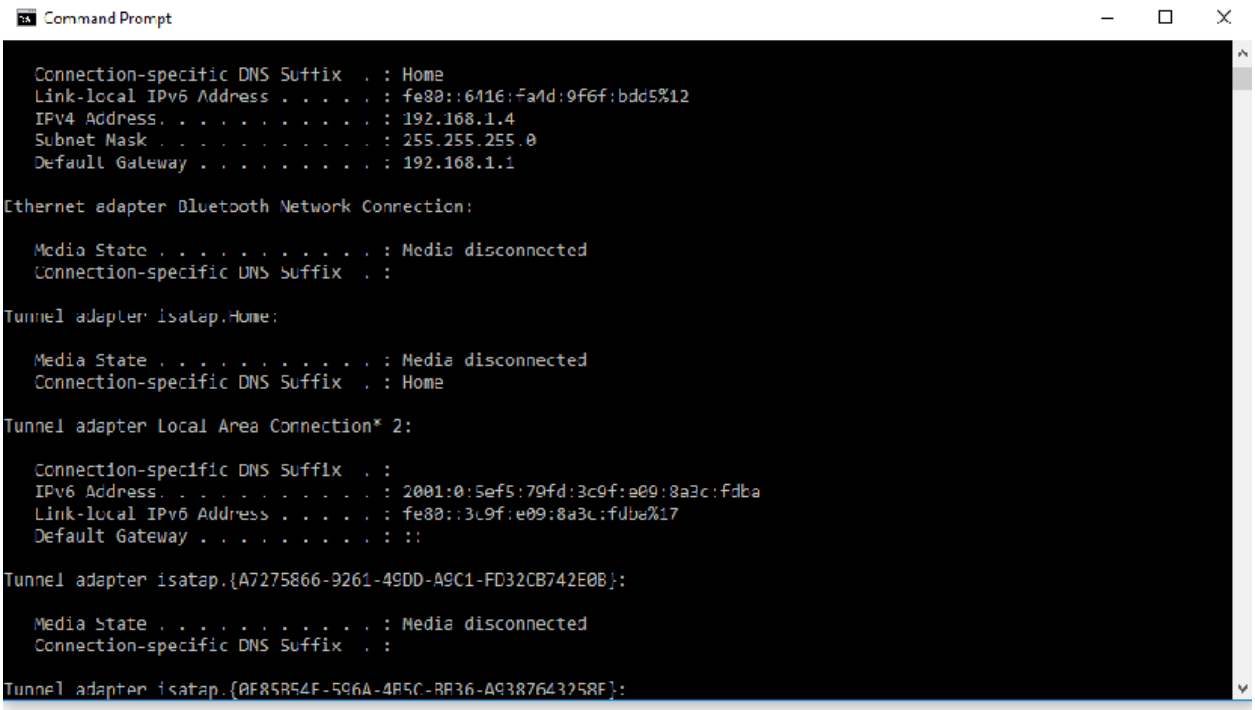


Image 10.7

The IP Address of the Attacker's system has been taken a note. It will help in proving that the attack is done from this system. The IP Address is obtained by using the command prompt. Command→ipconfig. It will give us the IP Address.

7.Flooding attack on the Proposed System

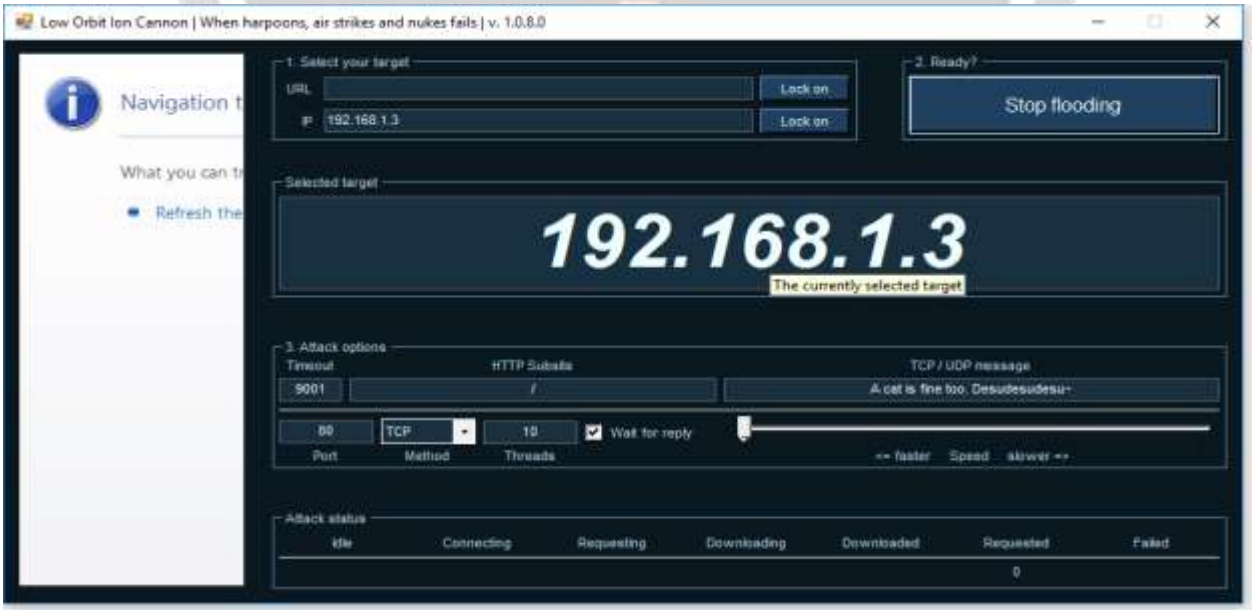
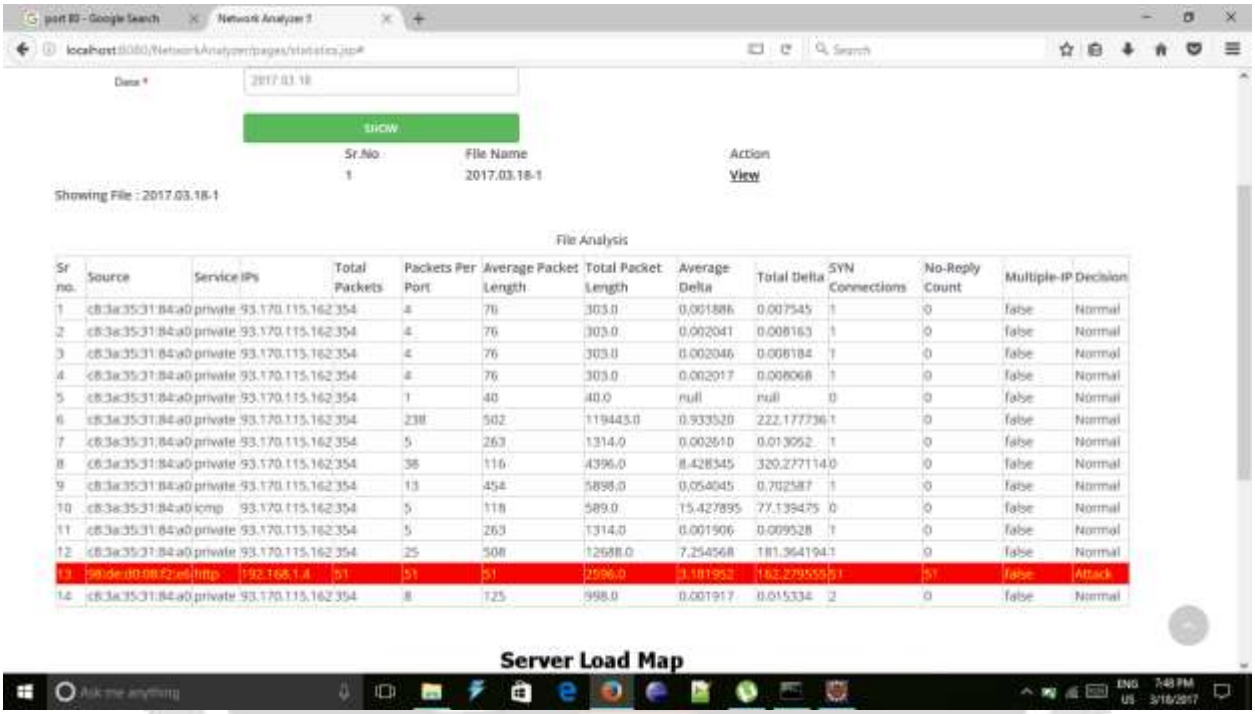


Image 10.8

Low Orbit Ion Cannon(LOIC) tool is used for flooding attack. The attacker uses this tool for performing flooding attack. The attacker just has to enter the IP Address of the system on which he wants to attack, then select the port and method i.e TCP,UDP. Then just press start flooding. Flooding Attack done by the attacker on the proposed system through the attacking tool on port 80.

8.Attack Detection



Sr no.	Source	Service	IPs	Total Packets	Packets Per Port	Average Packet Length	Total Packet Length	Average Delta	Total Delta	SYN Connections	No-Reply Count	Multiple-IP	Decision
1	c8:3a:35:31:84:a0	private	93.170.115.162	354	4	76	303.0	0.001886	0.007545	1	0	false	Normal
2	c8:3a:35:31:84:a0	private	93.170.115.162	354	4	76	303.0	0.002041	0.008163	1	0	false	Normal
3	c8:3a:35:31:84:a0	private	93.170.115.162	354	4	76	303.0	0.002046	0.008184	1	0	false	Normal
4	c8:3a:35:31:84:a0	private	93.170.115.162	354	4	76	303.0	0.002017	0.008068	1	0	false	Normal
5	c8:3a:35:31:84:a0	private	93.170.115.162	354	1	40	40.0	null	null	0	0	false	Normal
6	c8:3a:35:31:84:a0	private	93.170.115.162	354	238	502	119443.0	0.933520	222.177736	1	0	false	Normal
7	c8:3a:35:31:84:a0	private	93.170.115.162	354	5	263	1314.0	0.002610	0.013052	1	0	false	Normal
8	c8:3a:35:31:84:a0	private	93.170.115.162	354	38	116	4396.0	8.428345	320.2771140		0	false	Normal
9	c8:3a:35:31:84:a0	private	93.170.115.162	354	13	454	5898.0	0.054045	0.702587	1	0	false	Normal
10	c8:3a:35:31:84:a0	icmp	93.170.115.162	354	5	118	589.0	15.427895	77.139475	0	0	false	Normal
11	c8:3a:35:31:84:a0	private	93.170.115.162	354	5	263	1314.0	0.001906	0.009528	1	0	false	Normal
12	c8:3a:35:31:84:a0	private	93.170.115.162	354	25	508	12688.0	7.254568	181.364194	1	0	false	Normal
13	98:de:d0:08:f2:e6	http	192.168.1.4	51	51	51	2596.0	3.181952	162.279555	51	51	false	Attack
14	c8:3a:35:31:84:a0	private	93.170.115.162	354	8	125	998.0	0.001917	0.015334	2	0	false	Normal

Image 10.9

The proposed system shows that it has detected the attack done by the attacker. Above it is shown that the attack is made from 192.168.1.4.The proposed system has detected it as an attack and mentioned it in his decision.

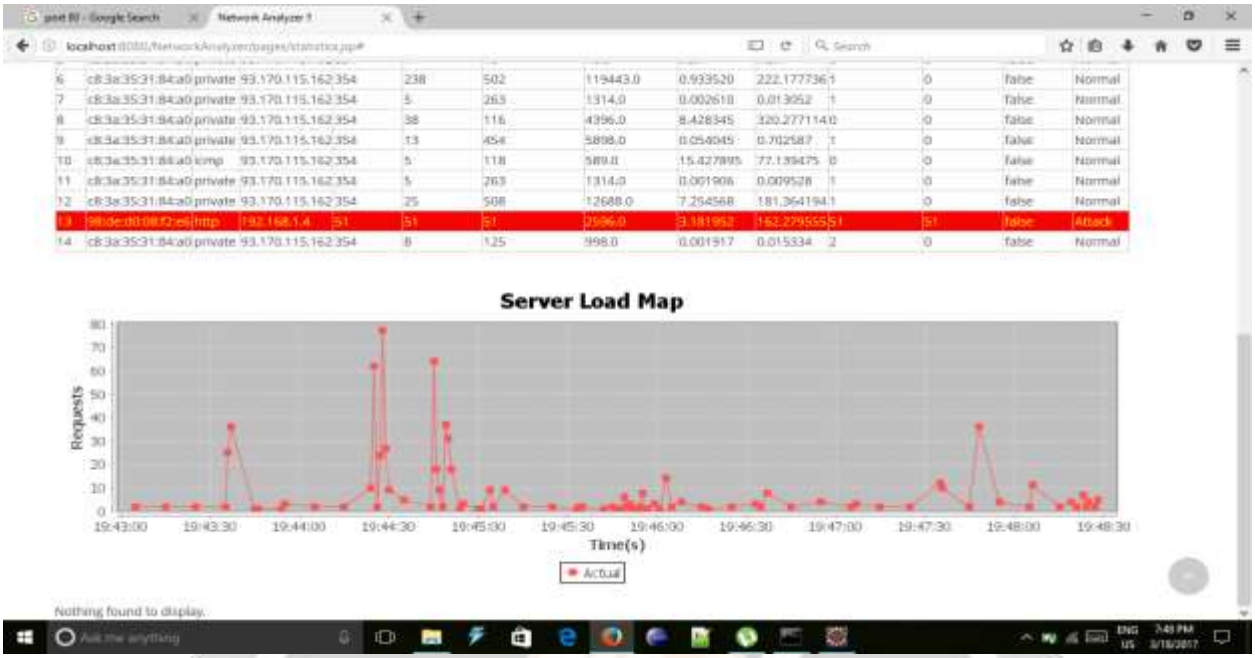


Image 10.10

The proposed system helps to detect the attacker and it also shows the load on the server at a time interval of 30 seconds with the help of the graphical representation. At the time the number of requests is high the graph reaches its peak and if low then it is normal.

3. CONCLUSIONS

We have shown through different sets of experiments that our proposed model addresses adequately two of the above challenges, namely, early detection and novelty detection. Our proposed model allows detecting bot activity in both the command and control and attack phases based on the observation of its network flow characteristics for specific time intervals. We emphasize the detection in the command and control phase because we would like to detect the presence of a bot early before any malicious activities can be performed, and we use the concept of time intervals to limit the duration we would have to observe any particular flow before we may raise our suspicions about the nature of the traffic.

4. REFERENCES

[1] HTTP-sCAN: Detecting HTTP-Flooding Attack by Modeling Multi-Features of Web Browsing Behavior from Noisy Web-Logs.
[2] Botnet Detecting Method Based on Activity Similarity.
[3] Botnet Behaviour Analysis using IP Flows with HTTP filters using classifiers.
[4] KOEHL A, WANG Haining. Surviving a Search Engine Overload. WWW 2012, April 1620, 2012, Lyon, France.
[5] RANJAN S, SWAMINATHAN R, et al. DDoS-Resilient scheduling to counter application layer attacks under imperfect detecting [J]. IEEE/ACM Trans. On Networks, 2009
[6] CNCERT/CC. China Internet Security Report 2012. POST TELECOM PRESS.Beijing, China: 2013
[7]]V.Kirubavathi and R.A. Nadarajan, ”HTTP botnet detection using adaptive learning rate multilayer feed-forward neural network,” in Information Securit Theory and Practice: security, privacy and trust in computing systems and ambient intelligent ecosystems.2012.
[8] C.Livadas, R. Walsh, D. Lapsley, and W.T. Strayer, ”Using machine learning techniques to identify botnet traffic,” in 2nd IEEE LCN workshop on network security, 2006.