A Real-Time (on Premise) Baseline Based DDoS Mitigation Scheme in a Hybrid Cloud

Ankur Rai and Rama Krishna Challa

Abstract Uninterrupted services are the most important factor for building customers trust towards a particular service providers, Distributed denial of service attacks are major threats towards disrupting the customer base for these service providers. Increasing sophistication of these attacks make them stealthier to evade existing perimeter security mechanisms. Hence, there is a need to design a dedicated mechanism to counter these attacks. In this paper we present a real time mitigation approach for DDoS attacks in a hybrid cloud. This approach utilizes a real time hybrid cloud test bed environment implemented with both intrusion detection system (IDS) and intrusion prevention system (IPS) components for result analysis and is utilized to mitigate signature based attacks at layers 3, 4 and 7 of TCP/IP network model. To implement this approach various stages to mitigate these attacks are considered. The results obtained have 100 % detection accuracy in all the scenarios considered.

Keyword DDoS · IDS · IPS · Mitigation · Dedicated mechanism · Hybrid cloud

1 Introduction

Denial of service poses a major threat towards an organization commercial success and also towards receiving unhindered access of services by the legitimate users. Distributed Denial of service (DDoS) attack is one such variation of this threat where large network of compromised systems called as zombies are utilized to perform these attack. These attacks are broadly categorized into three types [1]. First, Volume based attacks where the attackers overwhelm the victim with large

A. Rai (✉) · R.K. Challa
Department of Computer Science & Engineering, NITTTR, Sector 26,
Chandigarh 160019, India
e-mail: anks.rai@gmail.com

R.K. Challa
e-mail: rkc_97@yahoo.com

R.K. Choudhary et al. (eds.), *Advanced Computing and Communication Technologies*,
Advances in Intelligent Systems and Computing 452,
DOI 10.1007/978-981-10-1023-1_35

network traffic such that its resources and services are exhausted and are made unavailable to normal or legitimate users. Typical examples of such type of attacks are Ping flood and TCP-SYN flood attacks which occur at Layer 3 and Layer 4 of TCP/IP network model respectively. Second, Application attacks where various applications can be targeted by the attacker to exhaust the victim's resources. Typical example of such type of attacks is HTTP flood attacks. These attacks occur at Layer 7 of the TCP/IP network model. Third, Low rate attacks where the attacker exhaust the victim's server with minimal bandwidth and connections and hence are hard to detect. Typical example of such type of attack is Slowloris attacks.

2 Related Work

Defeating DDoS attacks [2] Explains the need of complementary solutions in spite of the existence of routers and perimeter security technologies to counter these attacks. They gave their own architecture to provide a complete protection form DDoS attacks. Authors in [3] proposed an approach by merging methods of data mining to detect and prevent DoS attacks, and used multi classification techniques like K-NN, Decision trees to achieve high level of accuracy and also to remove false alert alarms utilizing European Gaza Hospital (EGH) data set to detect the occurrence of DoS attacks.

In [4] the authors proposed an approach in which both the detection and prevention techniques are given. The detection is done utilizing the covariance matrix and TTL counting value and prevention is done using honey pot network. Authors in [5] presented an IDS that utilizes a layered framework integrated with neural network. This proposed system is shown to be more effective than other IDS models. In [6] an analytical approach to address the DDoS attacks with help of binomial distribution is proposed. The results show that the method effectively detects malicious traffic. In [7] a technique using special algorithm, CHARM is presented, which assigns a real–time risk score with the 2 way traffic. When there is an attack then it raises the threshold on the CHARM score and drops the highest risk traffic. Their technique is useful for detecting stealthy attacks. Authors in [8] implemented a virtual experimental setup and focused on three main types of application layer DDoS attacks, low rate, slow send and slow read. Their results show that it is difficult to attacks servers using slow rate and low rate due to their design and implementation. Devi and Yogesh [9] presents a hybrid technique against DDoS attacks based on the client trust value and the entropy. Here the trust value of clients is used to distinguish between the legitimate user and the illegitimate user. Authors in [10] presents an effective method is introduced to differentiate normal web traffic from application DDoS attackers using access matrix and Hidden Markov model. Ajoudanian and Ahmadi [11] explains data availability remains the most problematic area in cloud and gave a security model where security concerns and their solutions are categorized in three layers of security services. Anjali and Padmavathi [12] Explains that IPS devices and firewalls which allow application layer traffic are unable to distinguish between the normal traffic and attack traffic and an effective method is required to mitigate these attacks. In [13] various tools that can be utilized for testing purposes to perform application layer attacks are presented.

3 The Proposed Mitigation Scheme

The main objective of this paper is to develop a mitigation scheme in real time hybrid cloud environment for detecting and preventing signature based DDoS attacks at layers 3, 4 and 7 of TCP/IP model. The proposed scheme works by firstly, determining the attack profiles based upon variations in values of important network and system parameters. Secondly, setting the threshold values for these parameters using the attack profile generated. Finally, these threshold values and attack profiles are used as a baseline to detect and prevent any future DDoS attacks in the network. The proposed scheme works for three common types of attacks like Ping flood attacks, TCP SYN flood attacks, HTTP GET flood attacks. The proposed mitigation scheme follows a mitigation process for its testing and analysis which is shown in Fig. 1 and as explained below:

3.1 DDoS Mitigation Scheme

The mitigation scheme is divided into two phases. Intrusion detection system (IDS) consists of phase 1 and intrusion prevention system (IPS) consists of phase 2 and the entire mitigation process is divided into four steps as shown in Fig. 1. These

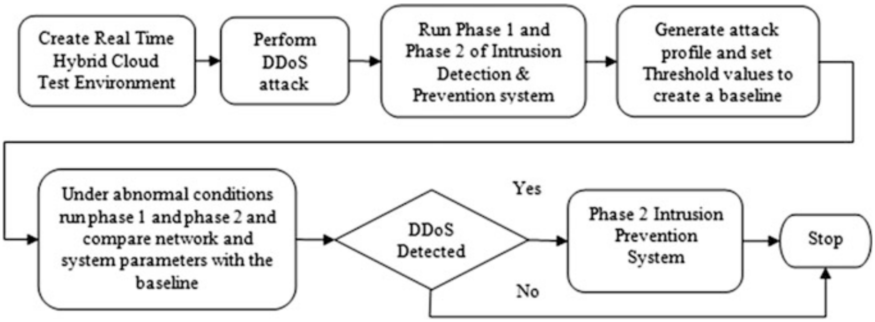

Create Real Time Hybrid Cloud Test Environment
Perform DDoS attack
Run Phase 1 and Phase 2 of Intrusion Detection & Prevention system
Generate attack profile and set Threshold values to create a baseline
Under abnormal conditions run phase 1 and phase 2 and compare network and system parameters with the baseline
DDoS Detected
Yes
No
Phase 2 Intrusion Prevention System
Stop

Fig. 1 Process for DDoS mitigation

steps are as explained as follows. In the first step, hybrid cloud environment is setup in the lab. This setup is implemented for testing and analysing the results. This setup provides both Storage as a Service (StaaS) and Software as a Service (SaaS) to the client machines. In the second step, DDoS attacks are performed on the implemented setup using existing DDoS attack tools. We have used three types of DDoS attack tools for performing attacks at the three layers (3, 4 and 7) of TCP/IP network model such as ping flood tool, hping3 tool, HULK (HTTP Unbearable Load King) tool respectively. Thirdly, phase 1 and phase 2 of the proposed mitigation scheme deployed at the server are applied simultaneously to detect the variations in the chosen system and network parameters and to detect number of incoming connections from different IP addresses respectively. The results obtained are used to generate attack profile and threshold values for important network and system parameters which are used to create a baseline for DDoS attack detection. Finally, whenever the system or network parameters show abnormal behavior like (low speed, unavailability of services, etc.) then phase 1 and phase 2 are run again to check the important network and system parameters values and compared with the baseline already created.

If the parameter values comply with the baseline then an attack is considered to be detected and phase 2 of proposed mitigation scheme is run to block the malicious IP addresses.

4 Experimental Results

Proposed IDS and IPS components of the mitigation scheme are implemented using Linux Shell Programming (BASH) and deployed on the system with Red Hat Enterprise Linux 6.4(64 bit) operating system, having Intel Core i7 Processor @ 3.4 GHz, 4 GB of RAM and 500 GB HDD. All the experiments are performed on the system with the mitigation scheme deployed.

4.1 Experimental Setup

The experimental setup for testing and analyzing the proposed DDoS mitigation scheme is shown in Fig. 2. This setup comprises of a cloud server deployed with the mitigation scheme and assigned a static IP address. Cloud server besides providing the cloud services is also configured to provide DHCP, MYSQL, HTTP services and a cloud server website developed and maintained at the cloud server. All the services can be accessed in both private (using Cisco Catalyst 2960 series network switch) and public cloud network except for the DHCP services which are available only for the clients in the private network. The Linux based client

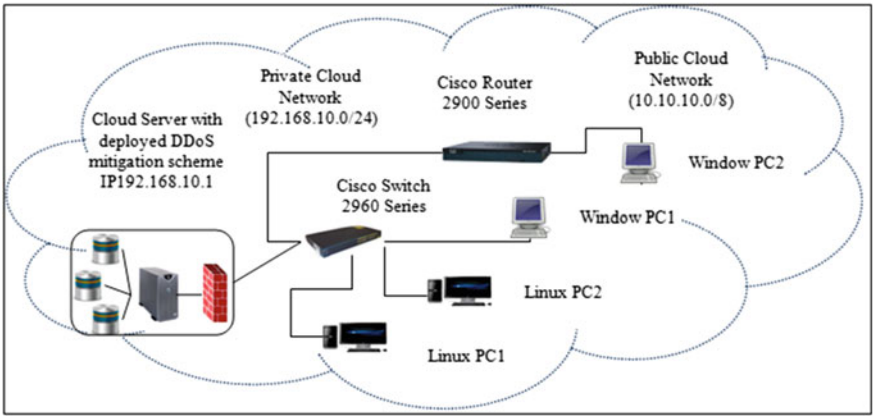


Fig. 2 Hybrid cloud environment setup for DDoS mitigation scheme

machines (Linux PC1 and PC2) with configuration depicted by Resource number 2 in Table 3 are used for performing the DDoS attacks like Ping flood, TCP SYN flood and HTTP GET flood on the cloud server. Windows machine (Windows PC1) with configuration depicted by Resource number 3 in Table 3 is used to determine the cloud server website load time during the DDoS attacks. Windows machine (Windows PC2) with configuration depicted by Resource number 3 in Table 3 and having a static IP address is used to access the cloud services in the public network via an intermediate router (Cisco Catalyst 2900 series). This router is configured with IP addresses of different network addresses (192.168.10.0/24 for private network and 10.10.10.0/8 for public network) on its ports. Next, we installed DDoS attack tools (hping3 and HULK) on the attacking machines (Linux PC1 and PC2) and Extended Status Bar tool for determining cloud server website load time on the Mozilla Firefox browser of Window PC1. After the setup is configured, we begin the process of generating the attack profiles and threshold values for creating a baseline for the DDoS attacks. In order to generate the attack profiles we considered important network and system parameters such as throughput, network latency, memory utilization, cpu utilization and socket utilization and website load time.

(1) Throughput
It is defined as number of bytes of packets transferred per unit time from source to destination. It is represented by Eq. (1) Here, TCP Receive window Size of 87380 bytes (Default value set on the cloud server) and average Round Trip Time are considered for throughput calculation

$$\text{Throughput} \leq \frac{\text{TCP Recieve Window Size}}{\text{Round Trip Time}} \tag{1}$$

(2) Memory Utilization
Memory utilization is represented by the Eq. (2). Here, Total Memory 4 GB is considered for calculations and other values are determined through system performance tool ‘vmstat’.

$$\text{Memory Utilization} = \frac{\text{Total Used Memory} - \text{Buffered} - \text{Cached}}{\text{Total Memory}} \times 100 \tag{2}$$

(3) CPU Utilization
It is represented by Eq. (3) and determined by subtracting the percentage of time CPU or CPU’S were in ideal or waiting state. I/O wait represent percentage of time CPU or CPU’S were idle during some I/O operation to take place, Steal represents the percentage of time spent in involuntary wait by virtual CPU or CPU’S while another virtual processor was being serviced by the hypervisor and Ideal represents the percentage of time CPU or CPU’S were ideal and there was no outstanding I/O disk request for the system and all these values are determined through system statistics tool ‘mpstat’.

$$\text{CPU Utilization} = 100\,\% - \text{I/O Wait} - \text{Steal} - \text{Ideal} \tag{3}$$

(4) Socket Utilization
In Linux, every socket when created returns a file descriptor [14]. Hence, socket utilization is determined based upon the percentage of file descriptors used during the attack. Here the default Max Open File Descriptor Limit of 374,382 set at the system is considered for calculations. Other values are determined through linux file descriptor commands such as (cat/proc/sys/fs/file-nr). This command gives both the values of the file descriptors (Max and Used). Socket utilization is represented by the Eq. (4)

$$\text{Socket Utilization} = \frac{\text{Total File Descriptors Used}}{\text{Max Open File Descriptor Limit}} \times 100 \tag{4}$$

(5) Network Latency
It is defined as time taken to send a particular request and receive an acknowledgement of that request. We determined network latency by sending ping packets to one of the client machines before and after the attack. Here only the average Round Trip Time is considered for determining this parameter.

(6) Website Load Time
It is defined as time taken to send a web request to the cloud server and time taken to receive the requested web page. Now, before generation of any type

of attack profiles and threshold values, our proposed mitigation scheme is run on the server to determine the initial values of all the parameters considered. This is done to check the variations during and after the attack.

4.2 Experimental Results of Ping Flood

Here, we run total 4 instances of Ping flood tool with a payload of 65,000 bytes on each attacking machine simultaneously. Figures 3, 4 and 5 represents results determined from Phase 1 and Fig. 6 represents results determined from Phase 2 of the mitigation scheme. The attack profile depicted from Figs. 3, 4 and 5 and the threshold values mentioned in Table 1. Represents a baseline for ping flood attack which is used to block the IP addresses (depicted in Fig. 6.) that are contributing maximum to the total received ping request. Figure 3 depicts the values of avg. network latency and avg. throughput before and after the attack. Here, the value of avg. network latency increased to 10719.686 % (62.105 ms) of its initial value (0.574 ms) and the values of Average Throughput, which decreased to 99.07 % (1.406 MB/s) from its initial value (152.229 MB/s) and Fig. 4 depicts the values of system resources (Memory, CPU, Socket) before and after the attack. The results show that these resources were affected mildly. Figure 5 depicts the profile of the website load times during the entire attack duration. The results show that denial of service reached within 50 min (∼1 h) at 3.45 PM and Fig. 6 depicts number of ping requests received from individual IP address and total ping request received during the entire duration of the attack. Based upon the profile depicted in Figs. 3, 4 and 5, threshold values for predicting ping flood attack are shown in Table 1. Both the attack profile and threshold values create a baseline for this attack.

Fig. 3 Average network latency and average throughput

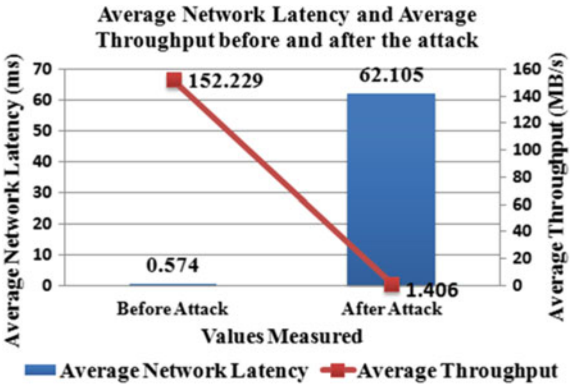


Fig. 4 System resources utilization

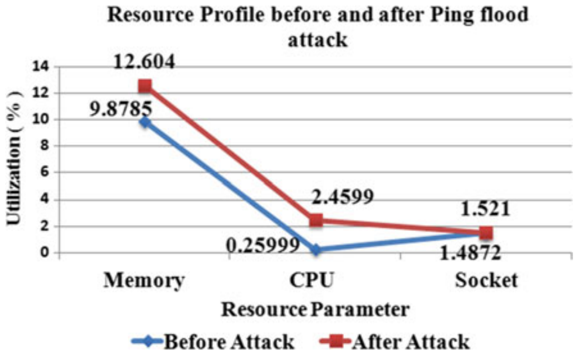


Fig. 5 Website Load time during the attack

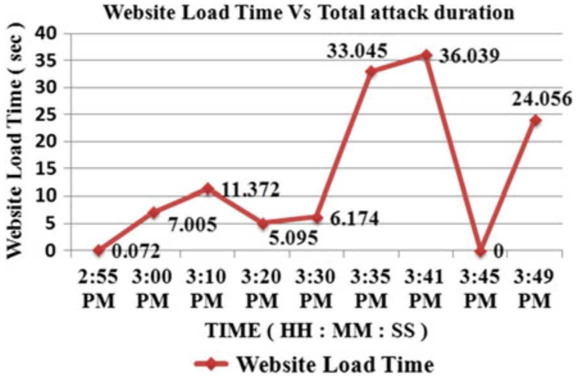


Table 1 Threshold values for ping flood attack

Parameter	Initial value	Final value	Threshold value
Avg. network latency	0.574 ms	62.105 ms	10719.686 % increase
Avg. throughput	152.229 MB/s	1.406 MB/s	99.07 % decrease
Memory utilization	9.8785 %	12.604 %	27.59 % increase
CPU utilization	0.2599 %	2.4599 %	846.15 % increase
Socket utilization	1.4872 %	1.521 %	2.272 % increase

4.3 Experimental Results of TCP SYN Flood

In order to generate profile for this attack, we noted the initial values of the parameters again and run one instance of hping3 tool on each attacking machine simultaneously. During the attack it was determined that this tool brought down the web services in 1 h. Figures 7, 8 and 9 represent the profile generated during the attack and Table 2 depicts the threshold values. Figure 7 represents the change in

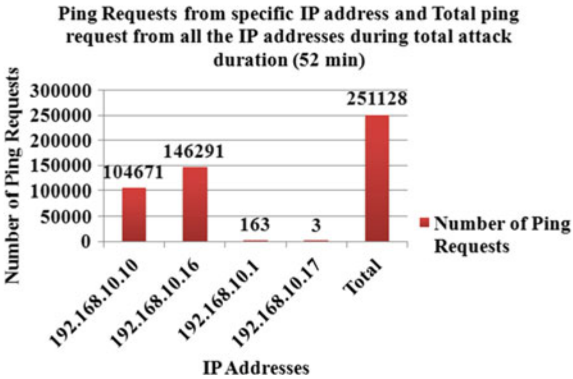


Fig. 6 Ping requests from different IP address and total ping request received

Table 2 Threshold values for TCP SYN flood attack

Parameter	Initial value	Final value	Threshold value
Avg. network latency	0.670 ms	8.352 ms	1146.56 % increase
Avg. throughput	130.4179 MB/s	10.4621 MB/s	91.9 % decrease
Memory utilization	9.8785 %	9.932 %	0.57 % increase
CPU utilization	0.3799 %	0.370 %	2.605 % decrease
Socket utilization	1.4701 %	1.478 %	0.53 % increase

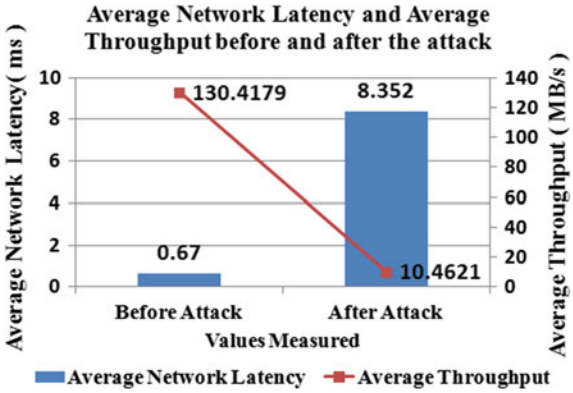


Fig. 7 Average network latency and average throughput

values of average network latency and average throughput before and after the attack. The results depicts that these parameters were affected more than other parameters and Fig. 8 depicts that system resources (memory, CPU and socket) were affected mildly. Figure 9 depicts the website load time during entire attack duration. The results show that denial of service reached in 1 h at 1.38 PM and Fig. 10 depicts number of TCP SYN request received from individual IP address and total TCP SYN request received during the entire attack duration.

Fig. 8 System resources utilization

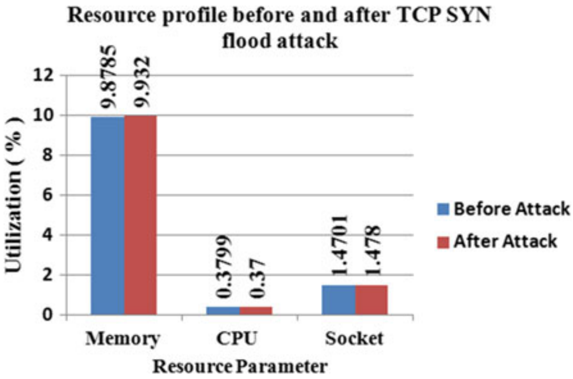


Fig. 9 Website load time

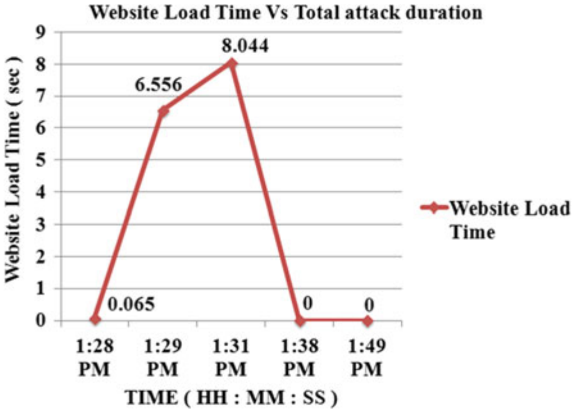
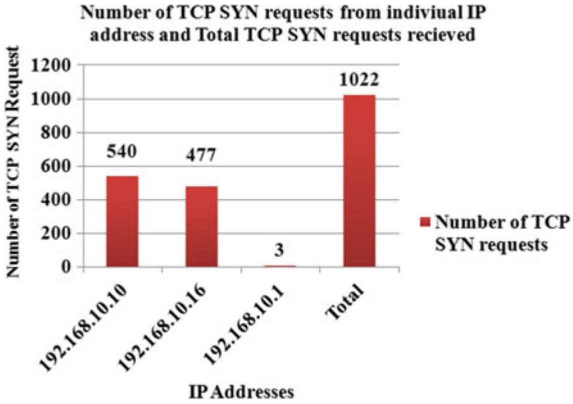


Fig. 10 Number of TCP SYN requests received



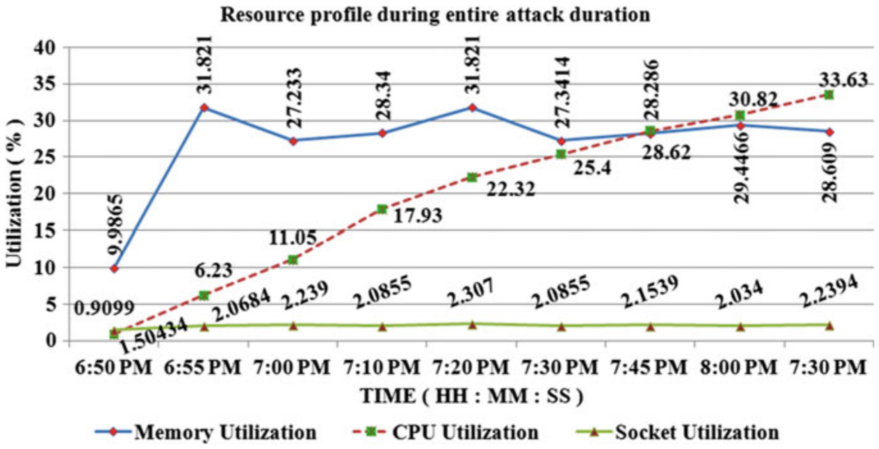


Fig. 11 Resource profile on i7 processor

4.4 Experimental Results of HTTP Flood

HTTP flood attacks traffic being slow-rate and low-rate appears to be as a legitimate traffic and hence are hard to detect. The results from these attacks revealed that stronger servers are affected less than servers with weaker configurations.

The result are determined using server configuration depicted by Resource number 1 in Table 3 with 8 CPU'S used (i7 processor) and with 4 CPU'S used (i4 processor). Figure 11 depicts the resource profile generated with i7 processor. Here the resources are constantly consumed depending upon the connection limit set on the server which can be modified depending upon the load on the server. However, this limit cannot be increased beyond the compile time connection limits. On, the server's where the compile time limits can be increased, these resource utilization can be 100 % and can lead to denial of service of resources and Fig. 12 depicts the resource profile with i4 processor. Here, the maximum memory, CPU, socket consumption reached to 58.704, 45.44, 3.446 % which were 31.821, 33.63, 2.2394 % respectively on i7 processor. Figure 13 depicts the results from phase 2 on i7 processor. Here, number of server sockets kept busy by individual IP address and total number of sockets busy at particular time is shown. At any particular time, IP address contributing maximum to the attack are considered as malicious and are blocked running phase 2 of the mitigation scheme. Similar results are obtained for i4 processor

In order to create a baseline for this attack only the attack profile is considered because Layer 7 attacks such as (HTTP-GET, HTTP-POST flood) are designed to work under the threshold limits. Hence, in order to counter such attacks only attack profile would be an effective solution.

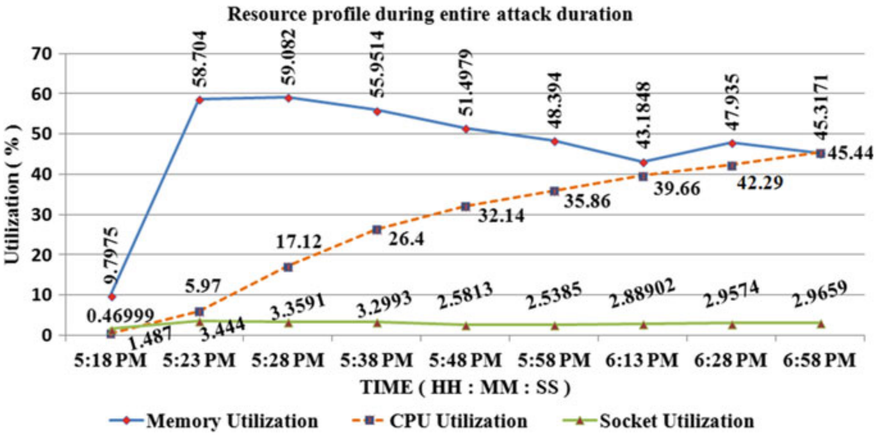


Fig. 12 Resource profile on i4 processor

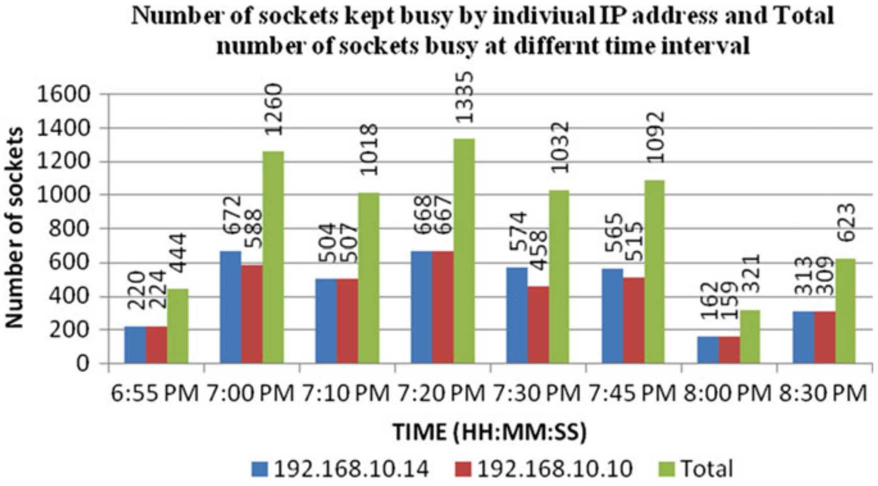


Fig. 13 States of sockets on i7 processor

5 Conclusion

In this paper, we present a real-time approach for DDoS mitigation, with both IDS and IPS components implemented. The proposed scheme besides providing an effective baseline based solution to counter signature based DDoS and DoS attacks can also be utilized for differentiating malicious traffic from legitimate traffic. The results obtained show high accuracy in detecting and preventing malicious IP addresses. The proposed scheme can be deployed in real organizational network. In the future we plan to detect botnet based DDoS attacks using this scheme.

Appendix 1

See Table 3.

Table 3 Resource/Device configuration

Resource number	Resource name	Configuration
1	Server (with File Descriptor Limit (FDL), ServerLimit (SL), and MaxClient limit (MC))	Intel core i7/3.4 GHz, RHEL 6.4(64-bit), kernel 2.6.42–358.el6.x86_64, RAM 4 GB, 500 GB HDD 32 bit or 64-bit, FDL-4096 (default: 1024) SL-200000 (default: 256), MC 200000 (def.256)
2	Linux PC1&PC2	Intel core i7/3.4 GHz, RHEL 6.4(64-bit), kernel 2.6.42–358.el6.x86_64, RAM 4 GB, 500 GB HDD 32 bit or 64-bit
3	Window PC1&PC2	Intel core i3/3.0 GHz, Windows vista (32 bit) RAM 2 GB, 150 GB HDD, 32 bit or 64-bit

References

1. DDoS Defense guide.: http://www.cisco.com. Accessed 6 Oct 2015
2. Defeating DDOS attacks.: http://www.cisco.com/c/en/us/products/collateral/security/traffic-anomaly-detector-xt5600a/prod_white_paper0900aecd8011e927.html. Accessed 6 Oct 2015
3. Tabash, M., Barhoom, T.: An approach for detecting and preventing DOS attacks in LAN. Int. J. Comput. Trends Technol. **18**, 265–271 (2014)
4. Ismail, M.N., Aborujilah, A., Musa, S., Shahzad, A.: New framework to detect and prevent denial of service attack in cloud computing environment. Int. J. Comput. Sci. and Secur. **6**, 226–237 (2012)
5. Srivastav, N., Challa, R.K.: Novel intrusion detection system integrating layered framework with neural network. In: 3rd IEEE International Advance Computing Conference, pp. 682–689 (2013)
6. Shyamala Devi, V., Umarani, R.D.: Analytical approach for mitigation and prevention of DDoS attack using binomial theorem with bloom filter an overlay network traffic. Int. J. Adv. Res. Comput. Commun. Eng. **2**, 3031–3036 (2013)
7. Defending against application-layer DDoS attacks.: http://www.juniper.net/assets/us/en/local/pdf/whitepapers/2000550-en.pdf. Accessed 8 Oct 2015
8. The Impact of Application Layer Denial of Service Attacks.: http://www.cs.unb.ca/~natalia/ApplDDoS.pdf. Accessed 8 Oct 2015
9. Devi, S.R., Yogesh, P.: A hybrid approach to counter application layer DDoS attacks. Int. J. Crypt. Inform. Secur. **2**, 45–52 (2012)

10. Rajesh, S.: Protection from application layer DDoS attacks for popular websites. Int. J. Elect. Eng. **5**, 555–558 (2013)
11. Ajoudanian, S., Ahmadi, M.R.: A novel data security model for cloud computing. Int. J. Eng. Technol. **4**, 326–329 (2012)
12. Anjali, M., Padmavathi, B.: Survey on application layer DDoS attacks. Int. J. Comput. Sci. Inform. Technol. **5**, 6929–6931 (2014)
13. Layer Seven DDoS Attacks.: http://resources.infosecinstitute.com/layer-seven-ddos-attacks/. Accessed 12 Oct 2015
14. Socket Programming. home.iitk.ac.in/ ~ chebrolu/scourse/slides/sockets-tutorial.pdf. Accessed 12 Oct 2015