International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-7, Issue-6, March 2019

Early Detection of DDoS Attack on Cloud Environment using Queuing Model

A.Saravanan, S.Sathya bama

Abstract— ***Due to the growth of the internet and related technologies, cloud computing plays the most significant part in providing cost effective services to the user. As the need for the cloud increases, the security issues related to the cloud environment is also increasing dramatically. The main challenge of the cloud environment is in providing quality service, data availability and managing the resources. The intensity of this challenge is increased due to the interruption of the Distributed Denial of Service attack, a most severe vulnerability that causes harm to the cloud environment. Though the attack is not a new risk for the research community, it takes a new dimension in providing a solution for the cloud environment due to its architecture and severe consequences. Due to the growing popularity of cloud computing, the mitigation of various vulnerabilities especially Distributed Denial of Service attack become the ongoing research challenge. In this paper, a framework that prevents and detects the attack at an early stage has been suggested to maintain the availability of cloud resources to its end users. The framework employs screening tests for preventing the cloud environment from Distributed Denial of Service attack. Additionally, the detection algorithm has been suggested that uses a queuing model for detecting the attack. The experimental results show that the proposed method provides a high detection rate.***

Keywords— ***Cloud Computing; Distributed Denial of Service; Detection; Queuing Model; Security Challenge.***

I. INTRODUCTION

Generally, Cloud computing is a pool of shared computer and hosted services that can be delivered over the internet with least effort. It provides a virtual solution to the users and they pay as their usage. In the past few years, several organizations have realized the importance of cloud and converted their business to cloud platform. The cloud allows the organization to concentrate on their business activities with its policy 'pay as you go' instead of focusing on other requirements such as computer resources and infrastructure. The main challenge of the cloud environment is providing on-demand services. The key features of the cloud are improved quality, scalability, resource availability and manageability. Apart from providing services, it offers the storage space for storing the users' data. This feature allows the user to access their data at anytime and from anywhere. The services provided by the cloud model includes software, platform and infrastructure to its users. This cloud model can be deployed as either a private, public, community or hybrid cloud in which the public cloud attracts maximum users as it is publically available to the user. Besides the benefits of the cloud computing, the major concern is the security breaches and potential risks involved in providing quality services to the users. As the data and resources are stored virtually on the remote place without transparency, the security takes new dimensions than traditional computing environment. Thus, security in the form of trustworthiness, service availability and sensitive data protection is the imperative concerns for cloud users. According to the survey by the International Data Corporation (IDC), security is the major barrier in using cloud environment [1]. A survey that highlights the concepts of cloud computing, security issues and the countermeasures are discussed in [2]. Distributed Denial of Service (DDoS) attack, a variant of Denial of Service (DoS) attack is the most common vulnerability that focuses on the distributed and cloud environment. The attack makes the resources unavailable for the end users by sending numerous service request from a group of hosts. The existence of the DDoS attack was recognized in June 1998 and this is considered as the first occurrence of DDoS attack in web history [3]. Inspite of great efforts dedicated by the security professions for controlling the attack, the attack continues to grow with higher impact [4]. The major motivation of DDoS attacks are analysed and are listed as avenge, extortion, political issues, testing the skill ability, and business competition between cloud providers [5, 6]. Generally, the attacker identifies the targeted system and launches the attack. The adversary makes the system busy by sending the enormous number of messages in such a way that handling those requests become difficult for the target system. Basically, the DDoS attack comprises two steps [7]. As a first step, the adversary identifies the target system to plant malware. This system is termed as bot master or botnet. The system identifies other vulnerable systems for planting the malware. These are normal systems and do not cause any harm until the malware is activated and are named as zombies. In the next step, the adversary identifies the vulnerable system to lunch the attack and signals the zombie to launch the attack on the victim. Thus the victim receives several attacks at the same point of time. Additionally, the zombies may not have the same malware, instead, each zombie will have been planted with various attacks. Thus the victim could not process the requests and the services provided by the system or server is denied.

Revised Manuscript Received on 30 March 2019.
***** Correspondence Author
A.Saravanan*, Department of MCA, Sree Saraswathi Thyagaraja College, Pollachi, Tamil Nadu, India – 642205
S.Sathya bama, 483, Lawley Road, Coimbatore, Tamil Nadu, India - 641003

If this attack continues for a longer period, it even excludes web spiders and web crawlers from visiting the website which indirectly decreases the page rank of the specific site. Because of which, the user may not show interest in visiting the site, since the pages are not shown by the search engines due to low ranking [8]. As the number of DDoS attacks incidents are increasing, the mitigation of the attack is most significant than any other security issues [9].

Thus, the paper focuses on detecting and analyzing Distributed Denial of Service (DDoS) attack in a cloud computing environment. The proposed framework can be used instead of image reCAPTCHA and other methods along with no CAPTCHA. The prevention strategies identify bots from humans through screening tests which can be performed after no CAPTCHA reCAPTCHA and the detection strategies mitigate the DDoS attacks such as SYN flood, ACK flood, Smurf attacks using queuing model. The performance of the system is compared with other methods and proves to be better with a high detection rate.

II. Related Work

According to the literature survey, the mitigation of DDoS attack can be employed at different stages. Somani et al. categorized the defense mechanism against DDoS as attack prevention, attack detection and attack recovery [10]. The authors analyzed the features of various preventive measures along with their pros and cons [11]. The preventive measure involves identifying and filtering the susceptible packets. As a DDoS attack prevention strategy, a challenge response system is used to distinguish the machine from the human. Most commonly used technique is Turing test named as CAPTCHA. Several variations of the test such as graphical test, text puzzles, crypto puzzles are suggested by the researchers [12-14]. However, several methods have the possibility of puzzle accumulation attack and even puzzle generation and space to store the images are additional overhead. Several noises such as waviness and horizontal stroke were also added to escalate the complexity of breaking the CAPTCHA with a computer program. The reCAPTCHA and image reCAPTCH which is an enhancement of CAPTCHA that supplies the websites with images of words that are hard to read for optical character recognition (OCR) software, as a challenge to the clients. Image identification CAPTCHA is also widely used in recent days. Various methods like Naming CAPTCHA and Anomaly Detection CAPTCHA are also in use. The main downsides of the above methods are the graphics generation and storage space overhead. Conversely, text puzzles can also be used to identify the bot system, but the limitation is OCR attacks. Apart from the challenge response system, other methods were also proposed to prevent the DDoS attack with restricted access [15, 16]. In the attack detection stage, the attack is detected before it causes harm to the system. Data mining techniques are highly influential in detecting DDoS attacks. A multi-measure multi-weight ranking approach was proposed to recognize the various substantial network attributes. This method makes use of wrapper, filter and clustering methods for detecting Dos and probe attacks by assigning weight to the network features [17]. Using the eminent machine learning technique called ensembles, the features are verified under four output filters. This multi-filter feature provides optimal result in identifying DDoS attack [18]. A variation of filtering techniques termed as threshold filtering which includes hop count [19] and request count [20] was suggested in the literature. A naive bayes classifier is employed to identify the attack from the normal traffic. This method involves two frequency based techniques such as discrete fourier transform and discrete wavelet transform for achieving the better result [21]. Another entropy based metric system was proposed for detecting both DDoS flooding attack whiac is a lightweight protocol [22]. A multiclassifier ensemble learning with hybrid heterogeneous properties and heuristic detection algorithm based on Singular Value Decomposition (SVD) was introduced to identify DDoS [23]. A neural network technique is used to train the system in detecting and filtering DDOS attacks [24]. A defense mechanism through pushback and resource regulation was introduced in mitigating the effect of the attacks [25]. A hybrid intrusion detection system (H-IDS), was suggested by Cepheli et al. [26] in perceiving DDoS attacks. This method integrates both the anomaly-based and signature-based detectors for achieving higher accuracy rates. A method was introduced by the researchers for detecting not only DDoS attacks but also constant attack and pulsing attack using the difference in the packet size. This method detects the attack from the legitimate traffic [27].

Like data mining, the statistical measures such as correlation, entropy and information gain achieves more importance in detecting the anomalies from the legitimate network traffic [28]. The statistical measures involve selecting and extracting the features from the network traffic. Various methods of feature selection and feature extraction are presented [29, 30]. A novel correlation measure was employed in detecting DDoS attacks [31]. An entropy-based approach was suggested and proved to be efficient in detecting modern botnet-like malware based on anomalous patterns in a network [32]. Recently, a protection policy to dynamically install security applications across the controller and switches has been proposed by Han et al. that identifies the accurate location of the botnet [33]. Support Vector Machine (SVM) algorithm has been used in creating the model for classification of DoS attacks and normal network behaviors [34]. The survey on the various distributed denial-of-service attack, prevention, and mitigation techniques was discussed in Mahjabin et. al. [35]. A simple distance based measure was introduced in preventing and detecting flood based DDoS attack [36]. The final stage of mitigating DDoS attack is a recovery stage in which the methods are to implement in the victim server to recover from the attacks. Several methods including migration and backup resources were suggested by a few researchers [37, 38]. But at this stage, the implementation of any method leads to an overhead of reserved resources and costs to the victim server. Thus, from the literature survey, it is clear that the main limitations in prevention strategies are graphics, image, puzzle and text generation as well as the storage space overhead.

In case of attack detection, few methods do not support more efficiency and scalability. In the last case of attack recovery, the reserved resources and the costs to the victim server are more.

III. PROPOSED SYSTEM ARCHITECTURE

The main goal of the system is to mitigate the DDoS attack by providing preventive and detective measures. The overall process of the proposed system is given below in Fig. 1. Queuing Model is used on the server side. A Finite Capacity Markovian Queuing Model M/M/1/K is used [39]. It is a single server queue with a queue size K. The server has two queues, Processing Queue and Waiting Queue. All the request from the client is verified and it is stored in the processing queue. The verification mechanism is given in subsection C.

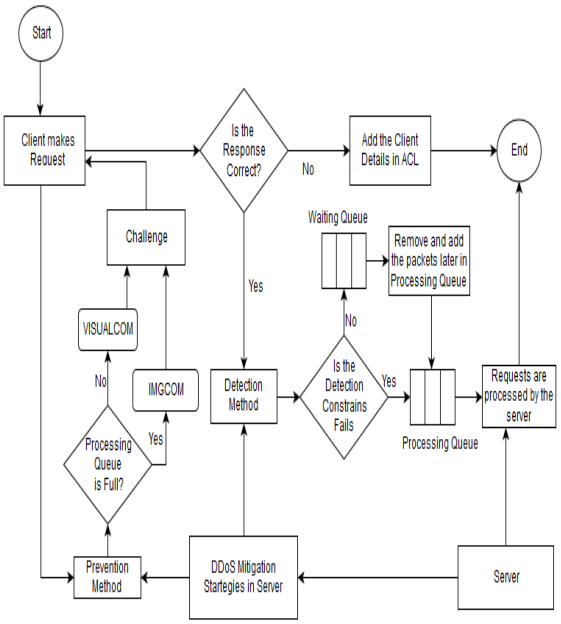

Fig. 1 The Overall Architecture of the Mitigation Framework for DDoS Vulnerabilities

The main goal of introducing the waiting queue is to process all the packets later, instead of dropping the suspicious packets. Also based on the number of packets in the queue, the prevention mechanism varies. The three enhanced CAPTCHA mechanisms are introduced namely VISUALCOM, IMGCOM, AD-IMGCOM. If the processing queue is full, then the intricate method IMGCOP or AD-IMGCOM is given as a challenge to the user, else the simple method VISUALCOM is used. These methods are explained in subsection B.

A. Queuing Model

Two queues are proposed in this framework. All the packets or requests arrived at the server are checked with detection constraints which are explained in subsection C. Initially, the packets waiting to be processed are stored in a processing queue. However, if any of the constraints met or if the processing queue is full, then the requests are stored on the waiting queue. Later, the requests are moved to the processing Queue for further process. Thus, all the requests are processed and even a suspicious request is also processed but with some delay or when the server is idle. The processing queue is finite with Poisson arrival and exponential service and the waiting queue is infinite.

The conventions of the queuing model are explained. The servicing method can accept and store k requests merely. The arrival rate of requests (the number of requests arrived per unit time) is denoted by λ. The effective arrival of the request (the rate of requests entering in the system) is given by λ_e. These requests are stored in the queue. k is the maximum number of services in the system. The rate of the request not entering the queue is given by λ_b. However, these requests are stored in the waiting queue. Thus $\lambda = \lambda_e + \lambda_b$.

The service rate of the requests (the number of requests serviced per unit time) is denoted by μ. An average number of requests in the system is given by L_s. The number of requests in the queue is given by L_q. The average waiting time of the request before completion of its request is given by W_s. The requests are processed in First Come First Served scheduling. In the queue, the requests wait for a time W_q for the service.

ρ is the utilization factor which determines the proportion of time that the server is busy servicing requests [40]. $\rho = \lambda/\mu$

Since it is a finite queue, the sum of probabilities till k^{th} request will be 1.

$$\sum_{n=0}^{k} P_k = 1$$

$$\sum_{n=0}^{k} (\lambda/\mu)^n P_0 = 1$$

$$P_0 = \frac{1}{\sum_{n=0}^{k} (\lambda/\mu)^n}$$

where $\sum_{n=0}^{k} (\lambda/\mu)^n$ represents finite Geometric Progression series.

Hence, the probability of no requests in the queue is

$P_0 = (1- \rho) / (1- \rho_{k+1})$

The probability of k customers in the system is given by

$P_n = \rho_n (1- \rho) / (1- \rho_k)$

If the arrival rate is lesser than the service rate, then the Processing Queue is said to be in steady state, since, all packets are serviced as fast as they arrive with minimum waiting time. Conversely, if the arrival rate is greater than the service rate then the system is said to be in an unsteady state since the packets must wait in a queue for a long time to be serviced.

The effective arrival rate is $\lambda_e = \lambda - [1-P_k]$

The average number of requests in the system $L_s = \sum_{n=0}^{k} n * P_n$

The number of requests in the queue $L_q = L_s - (1 - P_0)$

Using Little's law, the average waiting time of a request in the system can be determined as $W_s = \frac{L_q}{\lambda(1-P_k)} + \frac{1}{\mu}$

The average waiting time of a request in the queue is given by $W_q = W_s - \frac{1}{\mu}$

B. Prevention Strategy

The section presents the strategy for preventing DDoS attack in a cloud environment using the screening tests introduced in our previous work [41] to distinguish the bots from a human.

The three screening tests are named VISUALCOM, IMGCOM, and AD-IMGCOM.

The simple Visual comprehension (VISUALCOM) test was implemented by providing the visual scene or picture to the user. With this visual, a set of questions related to the visual scene is given as a challenge to the user. The user response with the answer to the respective questions. The main benefit of the proposed VISUALCOM screening test is the storage space required to store the images. In general, for Graphical Turing Test, each test needs one or more images for detecting the bots from the human which dramatically escalates the space complexity. Thus 'n' users need 'n' or more than n images to undergo a single Turing Test. Whereas, in the proposed VISUALCOM, with a single image approximately four to five questions can be framed. The second screening test is Image Completion (IMGCOP) which is a little complex test than VISUALCOM. The basic idea is splitting the image into various equal parts and the incomplete image is given as a challenge for the user. The user generally completes the image by dragging and dropping the various given image parts. If the completed image matches with the original image, then the response given by the user is authenticated. Similar to VISUALCOM, IMGCOM uses a single image as a challenge. The single image can serve various users by shuffling the remaining parts and providing the various combination of the incomplete image. The third method is similar to the Image Completion with the main difference is that the user is given with incomplete image and the remaining parts of the image with the anomaly part. This method is termed Image Completion with Anomaly Detection (AD-IMGCOM) and increases the complexity in such a way that the user must complete the image and identify the anomaly. The sample scenario is shown in Fig. 2.

Detection Strategy

Even after the challenge response verification, there is a possibility for flooding attack. Thus, this section provides the detection methods to identify the attacks. Each packet is analyzed to identify the attack. Several parameters are used to analyze the packets like IP address, TTL, etc.. Three variables Request Count CR, Source Count CS, SYN Flag Count CF are maintained by the server. The request count for each source is stored in CR. Similarly, new packets arriving continuously from different sources is stored in CS. Next, the packets arriving continuously with SYN flag on is maintained in CF to monitor SYN flood attack. Also, the threshold value for each variable is set to identify the DDoS attack.

The following are various conditions that are to be checked to move the packets to the processing queue and waiting queue.

1. A Time To Live (TTL) is an eight bit field to specify the maximum lifetime of an IP packet. From source to destination, the packet will pass through several routers and each router decrements the TTL value of an IP packet by one. As in [42] the server maintains IP2HC table. Each arrived packet is verified with the IP2HC table with the calculated HC and in Access Control List (ACL), if there is a match, then the packet undergoes the next security check, else it is spoofed. In such case, the packets can be added to the waiting queue and the ACL can be updated with the particular IP address.

2. If several packets arrived from the same source, then the count (Request Count CR) will be maintained. If the count exceeds the threshold value TR, then the packets are stored in the waiting queue else it undergoes the next check.

3. Similarly, if several new packets arrived continuously from different sources, again the count called Source Count CS, is maintained and updated. Again if it exceeds the threshold TS then the packets are moved to the waiting queue.

4. The next parameter is SYN flag. If several packets arrived continuously with SYN flag on, the count CF will be maintained. If the count exceeds the threshold TF then the packet will be moved to the waiting queue.

The packets stored in the waiting queue are processed after a minimum delay. The detection algorithm is shown in Fig 3.

C. Construction of IP2HC Table

The mapping between the IP address and the Hop Count is maintained by the server. The Border Gateway Protocol generally maintains the HC to other hosts for which it needs to communicate. Whenever it receives the packet from the particular IP address it verifies with the IP2HC table. If there is no match for the particular IP address, it broadcasts the Route Request RREQ packet to the neighbors with the particular IP address as the Destination Address (DA). On receiving the request, the neighbors verify the DA with its own IP address. If the destination address is not their own, they further forward to the neighboring routers.

The intended host, on receiving the request, sends the Route Reply RREP packet containing the route, hop count and other information to the source in the same route as RREQ but in reverse direction. Also, as in DMIPS [43], a verification step can be carried out by sending a query to the host and by setting the retrieved Hop Count as the TTL value. After the reply has arrived, the source then updates the IP2HC table with the IP address and Hop Count.

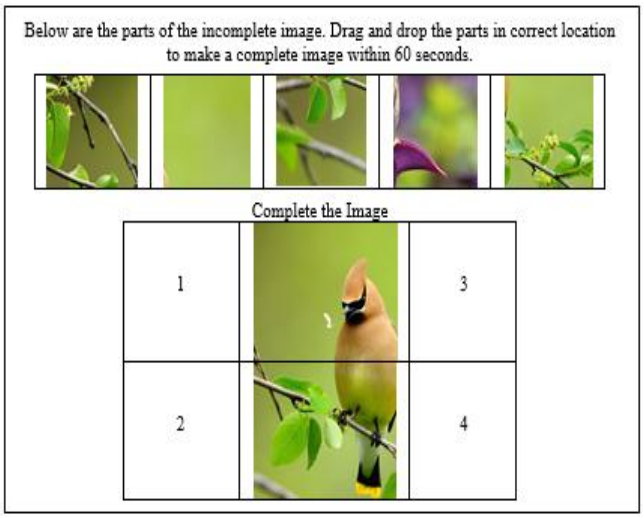

Fig. 2 Sample AD-IMGCOM Turing Test to identify the human and a bot with 5 parts and an incomplete image. The fourth part is an anomaly.

```
For each packet
Extract the IP address, TTL, SYN flag;
Update Request Count C_R, Source Count C_S, SYN
Flag Count C_F;
CH_1 = Compute HC from the TTL;
CH_2 = Access HC for the IP address from IP2HC;

If (CH_1 = CH_2) then
  Else if (C_R < T_R) then
    Else if (C_S < T_S) then
          Else if (C_F<T_F) then
            Store the packet in the Processing Queue
Else
            Store the packet in the Waiting Queue
   End If
End For
```

Fig. 3 Proposed Detection Algorithm

Once the packet has arrived, the IP address is matched with the IP2HC table. If there is a match, the HC will be calculated from the TTL value and compared with the Hop Count in the table referred as CH_2. The Hop Count value can be directly calculated from the TTL value of the received packet since the intermediate router decrements the TTL value of the packet before forwarding it to the next router. Since, the attacker cannot modify the values of the number of hops required for a packet to reach its destination, even though there is a possibility of modifying the fields in the IP header.

Hop Count CH_1= Initial TTL Value – Final TTL Value

The Final TTL Value is the one extracted from the received packet. The receiver calculates the Initial TTL Value which is more challenging. Luckily, maximum modern OSs employ only a limited and certain initial TTL values, 30, 32, 60, 64, 128, and 255. And since the maximum number of hops between any two nodes on the internet is more than 30 hops, the receiver can calculate the Initial TTL Value as the smallest value that is larger than the Final TTL Value. For example, if the Final TTL Value is 108, then the Initial TTL Value will be minimum of (128, 255) which is 128.

The proposed method detects various DDoS attacks in the cloud server. If several packets arrived from the same source, and if it exceeds the threshold value, the packets are moved to the waiting queue. This sets a limit for each client a configurable number of request to the server which detects layer 7 application layer DDoS attacks. Also if several new packets arrived continuously from different sources and the count exceeds the given threshold, the packets are moved to the waiting queue. This conditions will detect and mitigate the smurf attacks, ACK flood attacks. Generally, these two conditions detect layer 3, layer 4 and layer 7 DDoS attacks. The SYN flood attacks can be detected and mitigated by monitoring the count of the packets arriving with SYN flag on. Thus the proposed method outperforms in mitigating DDoS vulnerabilities.

IV. EXPERIMENTAL RESULTS

The evaluation of any detection method is extremely important before deployment in a real-time network. Thus the experimental analysis uses a single server and 15 clients to generate the network traffic and attack traffic. Several tools are available in the internet for simulating the attacks. Netwag tool is used in this work for generating well known DDoS attacks and JPCap tool is employed for capturing the the packets and in accessing all its header information. Generally, for transmitting and capturing the packets from the network, JPCap will be a perfect choice which is an open source java Library. The legitimate traffic has been used as a training data with 100 DDoS attacks. The overall prevention and detection rate for the proposed method along with the existing methods such as pushback [25], distance based [36] and C2DF [24] have been analyzed and the comparison is shown in Fig. 4.

The existing Pushback method is less effective since it uses high computation power and execution time. The distance based method does not perform effectively in case of recovery phase and resource utilization. The settings of C2DF method suffers from low accuracy in detecting the DDoS attacks. The graph in Fig. 5 shows the comparison of the false negative rate for all the existing methods and proposed a DDoS mitigation approach.

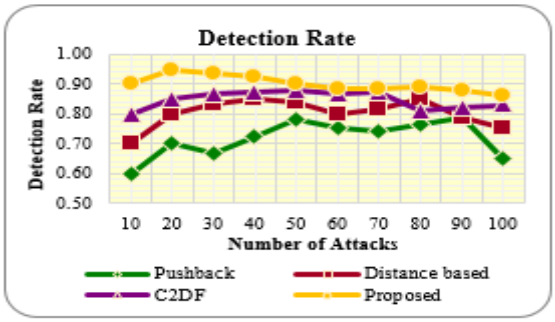

Fig. 4 Performance Analysis with DDoS Detection Rate

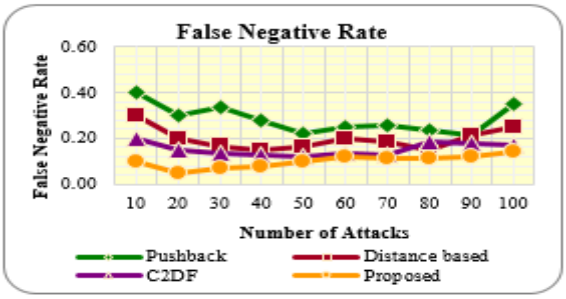

Fig. 5 Performance Analysis with DDoS False Negative Rate

The drop rate is the fraction of the number of packets dropped to the total number of packets. Fig. 6 represents the dropout rate comparison. From the analysis, it is clear that the dropout rate is minimum for the proposed mitigation framework when compared with the existing methods.

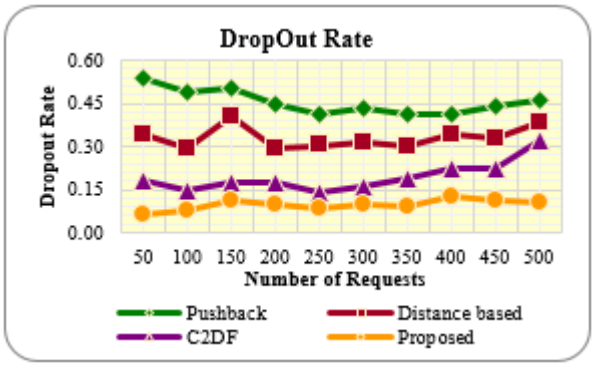

Fig. 6 Performance Analysis with DDoS Dropout Rate

Also, the performance evaluation has been made based on the execution time for the proposed model as a parameter. The time to fetch the images and to verify the response is analyzed for the proposed method. Table I shows the analysis of challenge response system with the execution time of three methods. The average execution time for VISUALCOM, IMGCOP and AD-IMGCOP are 29.2 ms, 42.6 ms and 48.6 ms respectively. Without the proposed prevention techniques, the execution time of the system is 14.4 ms. The average time delay is of 14.8, 28.2 and 34.2 milliseconds respectively, for the proposed methods which are negligible when the consequence of the DDoS attacks is considered.

TABLE I
COMPARISON OF EXECUTION TIME WITH AND WITHOUT PROPOSED PREVENTION METHODS

Test No	Execution Time in milliseconds			
	With VISUALCOM	With IMGCOP	With AD-IMGCOP	Without Prevention techniques
1	25	41	46	12
2	31	48	53	14
3	27	39	45	18
4	29	42	48	15
5	32	46	49	16
6	34	41	50	14
7	27	42	51	15
8	28	43	48	13
9	30	42	47	12
10	29	42	49	15
Average	29.2	42.6	48.6	14.4

Among the three prevention mechanism AD-IMGCOP takes more time and thus it is alone compared with other strategies such as reCAPTCHA and image reCAPTCHA. From Fig. 7, it is clear that the execution time is minimal when compared with the other existing techniques. Also, the proposed methods require less storage space when compared with the others and it is discussed at the end of the section. Thus if no CAPTCHA is not too sure, then the proposed method can be used instead of other existing techniques.

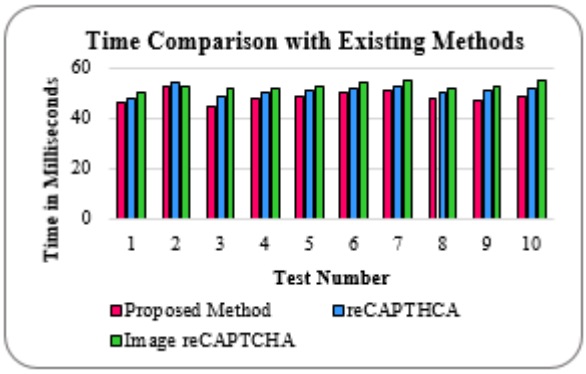

Fig.7 Execution Time Comparison of AD-IMGCOP with Existing Methods

To find out the performance of the processing queue, the analysis has been made by changing the size of the queue and the average waiting time of the system is calculated. The Arrival rate of packets is λ=9 packets/min. The average service time for a single packet is 10 seconds. Then the service rate μ=1/10 packets/sec=6 packets/min. thus the Utilization value ρ=9/6 = 1.5. The calculation of a number of requests in the queue, waiting time in the queue and the time taken to complete the request are given in Table II.

TABLE III
WAITING TIME AND COMPLETION TIME CALCULATION FOR VARIOUS QUEUE SIZE

Queue Size	Effective arrival rate (packets/min)	No of requests waiting in queue	Time to complete the service (in min)	Waiting time in the queue (in min)
5	5.8	3.4	0.76	0.59
10	5.9	8	1.5	1.3
15	5.9	13	2.3	2.1
20	6	18	3.1	3
25	6	23	4	3

Thus when the queue size is small, the performance of the system is better, since the waiting time is small for small queues than the long queues. The queue size is fixed at 10. The proposed framework has the capability to detect above 95% of attacks. However, when the threshold is decreased, the accuracy is increased even better. This is shown in Table III. The table shows the accuracy and error rate with the threshold value between 5 and 10.

TABLE IIIII
RESULT ANALYSIS WITH VARYING THRESHOLD VALUES

Threshold Value	Correctly classified attacks	Incorrectly classified attacks
10	95	5
9	95.4	4.6
8	96.2	3.8
7	97	3
6	97.8	2.2
5	98.6	1.4

The accuracy and error rate for the above implementation is shown graphically using the chart in Fig. 8. Also, the proposed method is analyzed with memory space as another parameter. For VISUALCOM, single image with 5 questions for each image is taken into account which can serve 5 users.

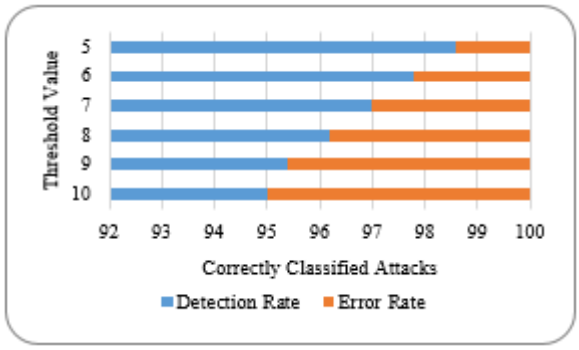

Fig. 8 Result Analysis with Varying Threshold Values

Similarly, IMGCOP stores a complete image with its part. In our experiment, each image is split into 6 parts and given as a challenge to the client. However, a single image can serve up to 15 users with different combinations. The third method AD-IMGCOP needs the same memory as IMGCOP along with small additional memory to store the anomaly image as well.

The existing method used for comparison is that an image reCAPTCHA in which a group of images that are selected based on the given challenge. This experimental result is shown in Fig.9.

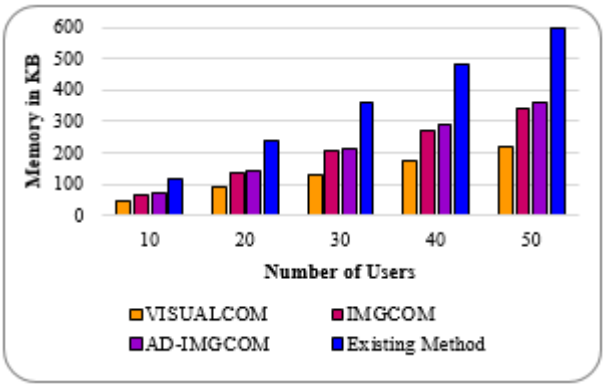

Fig. 9 Comparison of Memory Space in KiloBytes for Prevention Methods

Accordingly, the proposed methods require low memory space when compared to existing image reCAPTCHA. From the evaluation, it is clear that the prevention strategies proposed are stronger for the botnets to execute the DDoS attacks. Also, the memory requirement is minimum. The execution time is increased by the small amount which can be negligible

V. CONCLUSIONS

A defense framework for early detection of DDoS is discussed. An extensive literature survey shows that the existing methods are still struggling due to the attack in the cloud environment. The framework uses prevention and detection stages in which the prevention mechanism uses enhanced Graphical Turing tests to prevent the attacks from a botnet. In the detection stage, an algorithm that analyzes the packet header is added to the framework. This system is implemented with a queuing model. The experimental results are evaluated with a fixed queue size. The experimental results show the efficiency of the proposed framework. In the near future, a mechanism to modify the queue size dynamically can be introduced by evaluating the waiting time. Also, the future work aims at recovering the system from DDoS attack and other attacks that affect the cloud environment.

REFERENCES

1. Lonea, Alina Madalina, Daniela Elena Popescu, and Huanglory Tianfield. "Detecting DDoS attacks in cloud computing environment", International Journal of Computers Communications & Control, 8.1, pp: 70-78, 2013.
2. Ali, M., Khan, S.U. and Vasilakos, A.V., "Security in cloud computing: Opportunities and challenges", Information sciences, 305, pp.357-383, 2015.
3. Lin, Shun-Chieh, Shian-Shyong Tseng, "Constructing detection knowledge for DDoS intrusion tolerance," An International Journal of Expert Systems with applications, Vol. 27, No. 3, pp. 379-390, 2004.
4. Wang, B., Zheng, Y., Lou, W. and Hou, Y.T., "DDoS attack protection in the era of cloud computing and software-defined networking", Computer Networks, 81, pp.308-319, 2015.
5. Anwar, Z. and Malik, A.W., "Can a DDoS attack meltdown my data center? A simulation study and defense strategies", IEEE Communications Letters, 18(7), pp.1175-1178, 2014.
6. Osanaiye, O., Choo, K.K.R. and Dlodlo, M., "Distributed denial of service (DDoS) resilience in cloud: review and conceptual cloud DDoS mitigation framework", Journal of Network and Computer Applications, 67, pp.147-165, 2016.
7. Chen, K.Y., Junuthula, A.R., Siddhrau, I.K., Xu, Y. and Chao, H.J., "SDNShield: Towards more comprehensive defense against DDoS attacks on SDN control plane", In Proc. Of the conference on Communications and Network Security, pp. 28-36, IEEE, 2016.
8. N.Ch.S.N. Iyengar, Arindam Banerjee and Gopinath Ganapathy, "A Fuzzy Logic based Defense Mechanism against Distributed Denial of Service Attack in Cloud Computing Environment", International Journal of Communication Networks and Information Security, Vol. 6, No. 3, 2014.
9. S. Mansfield-Devine, "The growth and evolution of DDoS", Network Security, 10, pp. 13–20, 2015.
10. Gaurav Somani, Manoj Singh Gaur, and Dheeraj Sanghi, "DDoS/EDoS Attack in Cloud: Affecting Everyone out There!", In Proc. of the International Conference on Security of Information and Networks, ACM, New York, pp.169–176, 2015.
11. Somani, G., Gaur, M.S., Sanghi, D., Conti, M. and Buyya, R., "DDoS attacks in cloud computing: Issues, taxonomy, and future directions", Computer Communications, 107, pp.30-48, 2017.
12. VS Huang, Robert Huang, and Ming Chiang. A DDoS Mitigation System with Multi-stage Detection and Text-Based Turing Testing in Cloud Computing. In Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on. IEEE, pp.655–662, 2013.
13. Madarapu Naresh Kumar, P. Sujatha, Vamshi Kalva, Rohit Nagori, Anil Kumar Katukojwala, and Mukesh Kumar, "Mitigating Economic Denial of Sustainability (EDoS) in Cloud Computing Using In-cloud Scrubber Service", In Proc. of the International Conference on Computational Intelligence and Communication Networks, IEEE Computer Society, Washington, pp.535–539, 2012.
14. Fahd Al-Haidari, Mohammed H. Sqalli, and Khaled Salah, "Enhanced EDoS-Shield for Mitigating EDoS Attacks Originating from Spoofed IP Addresses", In Proc. of International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom, IEEE Computer Society, pp.1167–1174, 2012.
15. Bhavna Saini and Gaurav Somani, "Index Page Based EDoS Attacks in Infrastructure Cloud", In Recent Trends in Computer Networks and Distributed Systems Security, Springer, pp.382–395, 2014..
16. M. Masood, Z. Anwar, S.A. Raza, and M.A. Hur. EDoS Armor, "A cost effective economic denial of sustainability attack mitigation framework for e-commerce applications in cloud environments", In Multi Topic Conference (INMIC), pp. 37–42, 2013.
17. S Bhattacharya, S Selvakumar, "Multi-measure multi-weight ranking approach for the identification of the network features for the detection of DoS and Probe attacks", Compt. J. pp.1-21, 2015.
18. Osanaiye, O., Cai, H., Choo, K.K.R., Dehghantanha, A., Xu, Z. and Dlodlo, M., "Ensemble-based multi-filter feature selection method for DDoS detection in cloud computing", EURASIP Journal on Wireless Communications and Networking, p.130, 2016.
19. Tarun Karnwal, T Sivakumar, and G Aghila, "A comber approach to protect cloud computing against XML DDoS and HTTP DDoS attack", In Proc. of Conference on Electrical, Electronics and Computer Science (SCEECS), IEEE, pp.1–5, 2012.
20. Negi, Priyanka, Anupama Mishra, and B. B. Gupta, "Enhanced CBF packet filtering method to detect DDoS attack in cloud computing environment.", arXiv preprint arXiv:1304.7073, 2013.
21. Fouladi, R.F., Kayatas, C.E. and Anarim, E., "Frequency based DDoS attack detection approach using naive Bayes classification". In Proc. of international conference on Telecommunications and Signal Processing (TSP), IEEE, pp. 104-107, 2016.
22. Bhuyan, M.H., Bhattacharyya, D.K. and Kalita, J.K., "E - LDAT: a lightweight system for DDoS flooding attack detection and IP traceback using extended entropy metric", Security and Communication Networks, 9(16), pp.3251-3270, 2016.
23. Jia, B., Huang, X., Liu, R. and Ma, Y., "A DDoS attack detection method based on hybrid heterogeneous multiclassifier ensemble learning", Journal of Electrical and Computer Engineering, 2017.
24. Shamsolmoali, Pourya, M. Afshar Alam, and Ranjit Biswas, "C2DF: High Rate DDOS filtering method in Cloud Computing.", International Journal of Computer Network and Information Security 6.9 pp: 43, 2014.
25. Wang, X., "Mitigation of DDoS Attacks through Pushback and Resource Regulation", In International Conference on MultiMedia and Information Technology, IEEE, pp. 225-228, 2008.
26. Cepheli, Ö., Büyükçorak, S. and Karabulut Kurt, G., "Hybrid intrusion detection system for ddos attacks", Journal of Electrical and Computer Engineering, 2016.

27. Zhou, L., Liao, M., Yuan, C. and Zhang, H., "Low-Rate DDoS Attack Detection Using Expectation of Packet Size", Security and Communication Networks, 2017.
28. Wang, J. and Paschalidis, I.C., Statistical traffic anomaly detection in time-varying communication networks. IEEE Transactions on Control of Network Systems, 2(2), pp.100-111, 2015.
29. Hoque, N., Bhattacharyya, D.K. and Kalita, J.K., "FFSc: a novel measure for low‐rate and high‐rate DDoS attack detection using multivariate data analysis", Security and Communication Networks, 9(13), pp.2032-2041, 2016.
30. Hoque, N., Ahmed, H.A., Bhattacharyya, D.K. and Kalita, J.K., "A fuzzy mutual information-based feature selection method for classification", Fuzzy Information and Engineering, 8(3), pp.355-384, 2016.
31. Hoque, N., Kashyap, H. and Bhattacharyya, D.K., "Real-time DDoS attack detection using FPGA". Computer Communications, 110, pp.48-58, 2017.
32. Bereziński, Przemysław, Bartosz Jasiul, and Marcin Szpyrka. "An entropy-based network anomaly detection method." Entropy 17.4, pp: 2367-2408, 2015.
33. Han, B, Yang, X, Sun, Z, Huang, J and Su, J, "OverWatch: A Cross-Plane DDoS Attack Defense Framework with Collaborative Intelligence in SDN", Security and Communication Networks, 2018.
34. Mašetić, Z., Kečo, D., Doğru, N. and Hajdarević, K., "SYN Flood Attack Detection in Cloud Computing using Support Vector Machine", TEM Journal. Volume 6, Issue 4, Pages 752-759, 2017.
35. Mahjabin, T., Xiao, Y., Sun, G. and Jiang, W., "A survey of distributed denial-of-service attack, prevention, and mitigation techniques", International Journal of Distributed Sensor Networks, 13(12), P. 1-33, 2017.
36. Chapade, S.S., Pandey, K.U. and Bhade, D.S., "Securing cloud servers against flooding based DDoS attacks". In International Conference on Communication Systems and Network Technologies (CSNT), IEEE, pp. 524-528, 2013.
37. Jang, B., Doo, S., Lee, S. and Yoon, H., "Hybrid recovery-based intrusion tolerant system for practical cyber-defense". IEICE TRANSACTIONS on Information and Systems, 99(4), pp.1081-1091, 2016.
38. S. Yu, Y. Tian, S. Guo, and D. Wu., "Can We Beat DDoS Attacks in Clouds? Parallel and Distributed Systems", IEEE Transactions on PP, 99, 1–1, 2013.
39. Stewart, William J. Probability, "Markov chains, queues, and simulation: the mathematical basis of performance modelling". Princeton University Press, 2009.
40. Slothouber, Louis P. "A model of web server performance." Proceedings of the 5th International World wide web Conference, 1996.
41. A.Saravanan, M.S.Irfan Ahmed, S.Sathya Bama, "Mitigation Framework against DDoS Attacks in Cloud Server", ICT Innovations 2016 Web Proceedings, pp.33-42, 2017.
42. Doss, S., Narayanan, S. and Anand, J., "Detecting IP Spoofing using Hop Count Filtering based dynamic path update approach", Journal of Multidisciplinary Engineering Science Studies, 3(1), 2017.
43. Lagishetty, S, Sabbu, P & Srinathan, K, "DMIPS-Defensive Mechanism against IP Spoofing", Australasian Conference on Information Security and Privacy, pp. 276-291, 2011.