Mitigation Of Distributed Denial Of Service Attack Using Dynamic Captcha With Equal Probability Algorithm In Integration Of Internet Of Things And Cloud Environment

E.Helen Parimala, Dr.S. Albert Rabara, P.Theepalakshmi, Y.Sunil Raj

Abstract: Cloud Computing is developed to provide its users with access to resources worldwide as a human-centrated computing model. All power users who are specifically concerned with security in cloud computing may share resources. The attacks that cause attackers to challenge protection of CC include a denial of service and spreading a denial of service. Variety of defense systems have been developed and introduced in order to avoid denial of service attacks. Review of literature articulates that several proposed methods using simple Captcha test such as object pictures, distorted text, mathematical calculation and I m not a Robot is used for user authentication and secure integrated cloud environment and also avoid denial of distributed service attack, but existing methods has lot of limitations and it is not suitable for securing integrated cloud and IoT Environment. Hence, this paper proposes Dynamic Captcha Testing with Equal Probability Algorithm for first verification process, moreover puzzle is used for second verification process to strengthen security in Client side network furthermore to overcome distributed denial of service attack in the integration of Internet of Things and Cloud Environment.

Index Terms: Cloud Computing, Distributed Denial of Service, Dynamic Captcha Test, Puzzle, Firewall, Intrusion Prevention System, Internet of Things.

1. INTRODUCTION

The word IoT was first coined by a community member in development department of Radio Frequency Identification (RFID) called Kevin Ashton. Use of mobile devices for built-in communications, data analysis and cloud computing has resulted in a huge development [1]. IoT's Cyber Physical Systems (CPS), also referred to as IoT, comprises many smart systems and devices in the power, healthcare, and transport sectors in various ways, according to the definition from Smart America Global cities about IoT. Smart societies optimize the IoT software adaptation to enhance resilience and operational efficiency in improving life quality [2]. In the field of cloud computing services, the National Institute of Standards and Technologies (NIST) provided necessary basic aspects [3]. Cloud computing is a simple model that promotes access for its user network to the communicating system pool and requires no management effort and no reduced interaction between people or service providers.

Cloud computing is a modern computing technology that provides consumers with on-demand services and applications through the Internet. The way businesses are managed has changed, in particular for small and medium-sized firms [4].

- *Mrs. E. Helen Parimala, Assistant Professor, Department of Computer Science, M.V.M College, Dindigul, Tamilnadu, India. E-mail: helenandrew07@gmail.com*
- *Dr. S. Albert Rabara , Associate Professor, Department of Computer Science, St. Joseph's College (Autonomous), Tiruchirappalli, Tamilnadu, India. He has 21 years of experience in teaching and 13 years of experience in research. E-mail: a_rabara@yahoo.com*
- *Mrs. P. Theepalakshmi, Research Scholar, Department of Computer Science, National Institute of Technology, Tiruchirappalli, Tamilnadu, India. E-mail: theepalakshmirajan@gmail.com*
- *Mr. Y. Sunil Raj, Assistant Professor, Department of Computer Science, St. Joseph's College (Autonomous), Tiruchirappalli, Tamilnadu, India. E-mail: ysrsjccs@gmail.com)*

In fact, over the last several years the computing world has received much-deserved attention because of its characteristics that make it an attractive technology for many companies worldwide. Scalability, mobility, agility, on-demand, capability enhancement, elasticity, cooperation support and protection are the most important features in this respect [5]. In addition, elasticity provides a strong foundation for other advantages, including reducing overall costs and IT expenditures and consumers may profit from this innovation [6]. Businesses typically have their own servers and IT staff to operate and maintain these servers. The aim of these businesses is to provide services either to profit as a business company or to fulfill their mission as universities for their users [7]. In respect to cloud computing, these organizations, in addition to maintenance costs and IT employee salaries, can save expense of buying hardware, software licenses, and data storage, moving IT overheads to a cloud service provider. CSPs have huge resources which can be delivered in a flexible way on demand as a product. In addition, they have a significantly higher number of security professionals to protect their data centers and therefore customer information [8]. Despite the distinctive characteristics of cloud, its safety problems can hinder clients migrating to cloud. When clients don't have faith in the cloud, they won't accept it. Security considerations therefore constitute a good field for educational and industrial study [9]. The safety aspects can face several cloud risks. There are several security aspects. The list of risks contains the potential for violation of privacy, phishing chance, loss of information, and direct data control loss. However, several important aspects, which include the location of data, a disaster recovery plan, information separation, compliance with legislation, long term sustainability and research support, must be addressed prior to transitioning into the cloud [10]. Two different systems that are part of our daily lives are IoT and cloud computing. They are said to be the core part of the future internet, despite its continued use and adoption. Nonetheless, in some deployed software environments the combination of IoT with the cloud environment is expected to be a challenge.

The cloud and the IoT seek a new level of expertise in the assignment management that makes IT administration and business attractive for their profits. Because of its security problems, IoT and cloud combinations were not intended to be used in many applications and remain in the study of reading. A number of difficult cloud highlights are due to a lack of protection, confidentiality, QoS, reliability, denial of service etc [11]. Victims also focus on the development of IT that causes severe cyber-attacks on a daily basis. Several types of attacks occur in network security, which can damage network resources and services. Denial of service is one of the most common attacks in this regard. Denial of service (DoS). There is, however, a problem, which can cause customers to opt out [12]. A distributed denial of service (DDoS) attack involves a malicious attempt to disrupt the normal traffic of a targeted server, device or network by overwhelming the destination or its surrounding internet traffic infrastructure. DDoS attacks achieve productivity through the use as a vector of attack several compromised computer systems. Computers and other networked assets such as IoT devices can be included in managed machines. A DDoS attack on a high level is like a bottleneck blocked by the lane, preventing normal traffic from entering a desired destination [13]. In order to attack a DDoS, an attacker must gain control of a network of online machines. Computers and other machines (e.g. IoT devices) are malware-infected and each bot is transformed into a zombie. The attacker then manages the bots community remotely, which is known as a botnet [14]. Once a botnet is created, the attacker can manipulate the machines with the remote control method by sending updated instructions to each bot. Each bot responds by sending requests to the target, which may lead to overload of a targeted server or network, resulting in a denial of service to the normal traffic, when the IP address of a victim is attacked by a Botnet. Since every bot is a legitimate internet device, it can be difficult to separate attack traffic from normal traffic. To safeguard their functionality and the ability to provide services to their customers, shielding networks from malicious attacks is important. A good authentication mechanism [15] can be accomplished for external users, resource allocation based on their positions, and access control rule management. The network can be exposed to various threats, such as robbing sensitive data, if such procedures are not established. To overcome this distributed denial of service attack this paper presents Dynamic Captcha testing with the equal probability algorithm is used for in the first verification process in the client side network Cloud environment is an efficient way to provide security and also used to detect a DDoS attack. In addition, a contrast with the other DoS detection algorithm had also been done and it was found that the proposed mechanism had a more effective mechanism and a minimum attacker frequency. It is a proactive technique that checks user's legitimacy at the beginning of network access. The next section presents the literature review related to the work that motivates us in pursuing the work in this particular domain. Section 3 addressed the proposed architecture. Types of Dynamic captcha test is described in Section 4. The next section is carried out by the outline of the Dynamic captcha with equal probability algorithm Section 5 displayed experimental results. Conclusion is given in the following section.

2 REVIEW OF LITERATURE

In the DDOS attack detection in the cloud that is available in several papers, multiple research was conducted. On this segment, few of these papers are discussed. A method to identify DDoS attacks is suggested in paper [16]. The researchers develop a technique based on range estimates for traffic level determination. This distance calculation is used to determine the Time-To-Live (TTL). Even exponential IP traffic smoothing provides real-time measurement. Eventually, an abnormality is measured to determine whether or not the behaviour is normal. In this approach, time delays are used to resolve attribute dependency as demonstrated for previous approaches. Currently, ISPs are dependent on filters for data traffic, making it unlikely to pursue this strategy. The researchers, S.T Zargar et.al, suggested the Distributed, Collaborative, and Data-driven System for Detection and Prevention of Intrudents in [17]. Three (infrastructure, platform and software) levels are built on the framework. The clusters in the DCDIDP architecture communicate with three (Intrusion Assessment Information Base (IAIB), Policy and Regulatory Base, Audit Logs) regional databases to efficiently identify and avoid DoS and DDoS assaults. IAIB includes information on attack trends, standard packet actions, data access details. The policy and rule basis includes complex and political rules to allow users to manage their systems at various levels of architectureW. Dou et.al proposed a CBF approach for the security of cloud servers from DoS-and-DDoS-Aggression in [18]. Trust values are generated and stored on the basis of the attribute value pairs in the TCP and IP headers in the nominal profile. On the basis of the marginal profile, CBF determines scores for user's incoming data packets. The confidence values represent the occurrence rate of packets with a low score. When the packet score is high, the packages from that source are considered as legitimate packets and sent to the server for cloud access. M. Malekzadeh et.al suggested two different security models in [19]. The first design suggested is the PCF-O2, which is based on the original algorithm HMAC-SHA2-256 (O-hmac2). The KDA, new safety elements and replay attack protection scheme were suggested to be the Key Derivative Algorithm. For the second proposed model, the HMAC-SHA2-256, known as M-hmac2 is amended, but M-hmac2 is the foundation of second proposed authentication algorithm for the PCF-M2 template. The M-hmac2 was designed to reduce the O-hmac2 security risk and connectivity costs, thus enhancing and maximizing the efficiency of PCF-M2 versus PCF-O2 design. In [20] author R. Anandhi et al proposed to solve attacks by Service File Access (SFA) algorithm against DoS and DDoS on distributed port locking cloud servers. The main way to lock the port after the client accesses this solution. The Group Policy Object (GPO) was designed for users and files to control client end ports. The tripping code is created with a limited validation time when the user requests to access the folder. Since the port is closed, no other users have a chance of entering the port and if the user does not enter the code in the time frame, the user is considered to be a malicious user and is discarded. Writer [21] Rahman proposed to create an additional CC-defense layer against dos and dos attacks using hardware-based watermarks and filtering mechanism. The validity of the data is verified by hop count and TTL using a trace back method. If the authenticity is not verified, the package is labeled as a malicious packet and dropped without being reached by the network. If packets are accepted and marked as valid trace back packages, further review will be conducted for the "knowledge based server." If suspicious packets are found,

they will be marked and dropped as untrusted packets. Borean et al suggested [22] IDPS technique focusing on third party auditor (TPA) for CC against attacks by DoS and DDoS. Dempster Shafer Theory (DST) is used for this method. Three steps will be included in the proposed method. It is the identification, conversion and attack evaluation stage. Snort is used for detecting and logging flooded packet values in detection phase. During a conversion point, front server converts warnings to BPA based on the snort-generated assault alerts. In the analysis process, the converted bpa is melted and the attack is calculated on the basis of the normalized variable. To do this, it uses the combination principle from Dempseters. Patel et al [23] have suggested graphical password encryption, which includes three different picture classes, such as Famous Spot, Famously Men, Reputed Company Name each with 25 images. In order to protect user data or unauthorized data access, the author introduces captcha-based image. In this password images and password are created. Current system relies only on a password but has small password disadvantages which can mostly be remembered and used. This type of code can easily be imagined by various attacks, i.e. a dictionary and a brute force attack. Writer has offered a new password image Recognition technique is used using a numerical password to make text and graphic password safer and easier to remember. The three level CAPTCHA system was used by Rahman et al [24] to make cracking a difficult task for BOT. They noted that there is a relatively low probability that a BOT could hack the proposed prototype device. Huang, et al [25] used text-driven CAPTCHA for one of the purposes of minimizing the IaaS cloud service is DDoS (distributed denial of service). The challenge response system will reduce traffic and assess whether the packs are human or computer-based (program). There has been a great deal of work to provide a new or more stable Captcha format.

3 PROPOSED ARCHITECTURE

The proposed architecture is intended to inspect the origin (legitimate or malicious) of request, using dynamic Captcha test with equal probability algorithm for the first verification process. It is a proactive method which checks packet origin with dynamic Captcha test at the beginning of a connection to identify bots (infected machines) that attempt to access the system. The second verification follows by selecting a random packet sent from source for puzzle analysis. The firewall controls the defense system by assigning two primary lists: black and white with two secondary lists: provisional and permanent for package sources based on confirmation tests and track remaining packets using intrusion prevention system and reverse proxy server, to verify malware components.

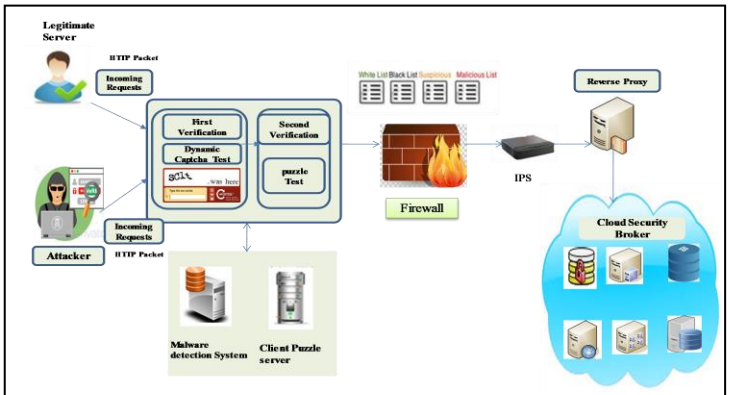

Figure 1 Architecture for Mitigation System of Distributed Denial of Service (MSOFDDOS)

This can be used to combat DDoS attaching two more key lists of suspicious or malicious lists of firewalls have been introduced to increase reaction to DDoS attacks. To prove its validity and efficiency, the suggested model was evaluated.

3.1. Types of Captcha Test used in Proposed Architecture

Security is the greatest problem today in our daily lives. Captcha is used to protect itself from various attacks and separate humans from robots. AltaVista developed CAPTCHAs originally to avoid submitting URLs to the search engine. It was a basic CAPTCHA that requires users to type a deflected word in English. Essentially, CAPTCHA is a completely automated user turnover control for machines and humans. Such systems are used with additional protection such as secure customer-to-authentication networks. CAPTCHA is given a more technical definition in [26]: "CAPTCHA is the crypto graphical protocol that is based on the assumption of hardness underlying the AI problem." For reality, CAPTCHA checks are used most often for online surveys, free email services, shopping agents, search engine bottlenecks, worms and spam, and dictionary protection. As a final step in registration process, e-mail service providers including Hotmail and Yahoo provide a CAPTCHA check that prevents bots from subscribing and using their spam tools. The Turing test is used in the artificial intelligence (AI) field to provide the information about a machine. Testing uses a method that places a human user and a computer in a number of rooms. The human interrogator can also ask questions in the third room. When the questioner is unable to identify the human or machine positions, this results in the device having passed the Turing test. CAPTCHA is a Turing test but is very different from the above description. If a machine rather than a person is substituted for the interrogator, it is called CAPTCHA. Human users can easily answer the request, but the existing computer programs were difficult or could not respond [27]. The relative importance of these features depends on the type of CAPTCHA. The following are the concepts driving CAPTCHA:

- A choppy image on which certain texts are projected is shown to user. Using random text to create this photo from database.
- In a text field displayed, the user shall enter same letters in the message.
- The server checks when uploading a form whether user's text matches text initially created. The payment persists if it does. A warning message is shown and a new code must be

entered by user otherwise.

• Exploits fact that many pattern recognition activities often offer people something more than machines.

3.1.1. Text Captcha

CarneGie Mellon has developed Gimpy which is to pick a dictionary word and ask users, after making a distorted text-containing image, to type what they see as an object. Yahoo uses this methods simple version EZGimpy. The image manipulation of EZ-Gimpy consists of ambient lines, gradients, non-linear deformations, whirling noises, pixel noise. Text of " stuvwxyz023456789" consists of the characters and amount. The text consists of five characters and has a bending value and size of each character. Most people can read from the distorted image three words, while existing computer programs do not. In addition,' W9XZq' is the text shown in Figure 2 below. The human but not the OCR system. This text is easily recognizable. That division has a random angle value between-3 and 3 degrees and is divided into eight (4x 2) pieces. Random width and height values are present in fragments. The CAPTCHA text and background are in similar colors.

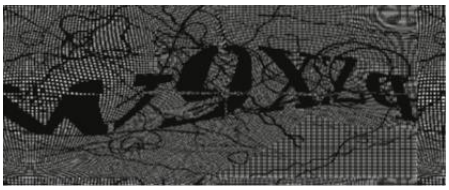

Figure 2 *W9XZq Example of Text Captcha*

Hotmail is a Microsoft Cooperation free email service, and a different method of CAPTCHA is used [28]. The random selection of a string of English characters is rendered and users are requested to type what they see after making those changes. This approach has the biggest drawback because of the curves between characters, some of the characters are read differently.

3.1.2 Image Captcha

The CAPTCHA image [29] is that pattern recognition is a serious AI challenge, which makes it difficult to crack this test by applying pattern recognition techniques. A broad labelled image server is used by Pix. It shows a set of images and the user needs to access the normal function.

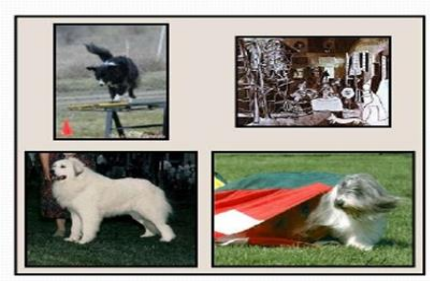

Figure 3 *Example of Image Captcha*

3.1.3 I m not a Robot Captcha

This is the latest iteration of the continuous war between spammers and computer scientists [30]. "I'm not a robot." This means a "comprehensively automated public turnover check to tell machines and people apart." Machines should not read them, although people can easily read them.

Figure 4 *Example I m not a robot Captcha*

3.1.4 Mathematical Calculation Captcha

Choose where to use math Captcha [31] account, register, missing password files, bbPress post, and Contact Form 7. Displaying Captcha as numbers and/or words for logged in users, choose which math procedure to use.

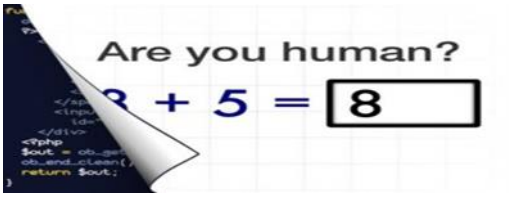

Figure 5 *Example Mathematical Captcha*

3.2 Dynamic Captcha with equal probability

An Automated Public Check CAPTCHA is an examination to find out if the consumer is human or not. Computers and humans are different from the test. The job is, therefore, to produce distinctive CAPTCHA each time, to determine whether or not the client is human being by asking the operator that he or she automatically enters the same CAPTCHA and to check the user input with CAPTCHA.

3.2.1. Captcha Verification Process Sequence

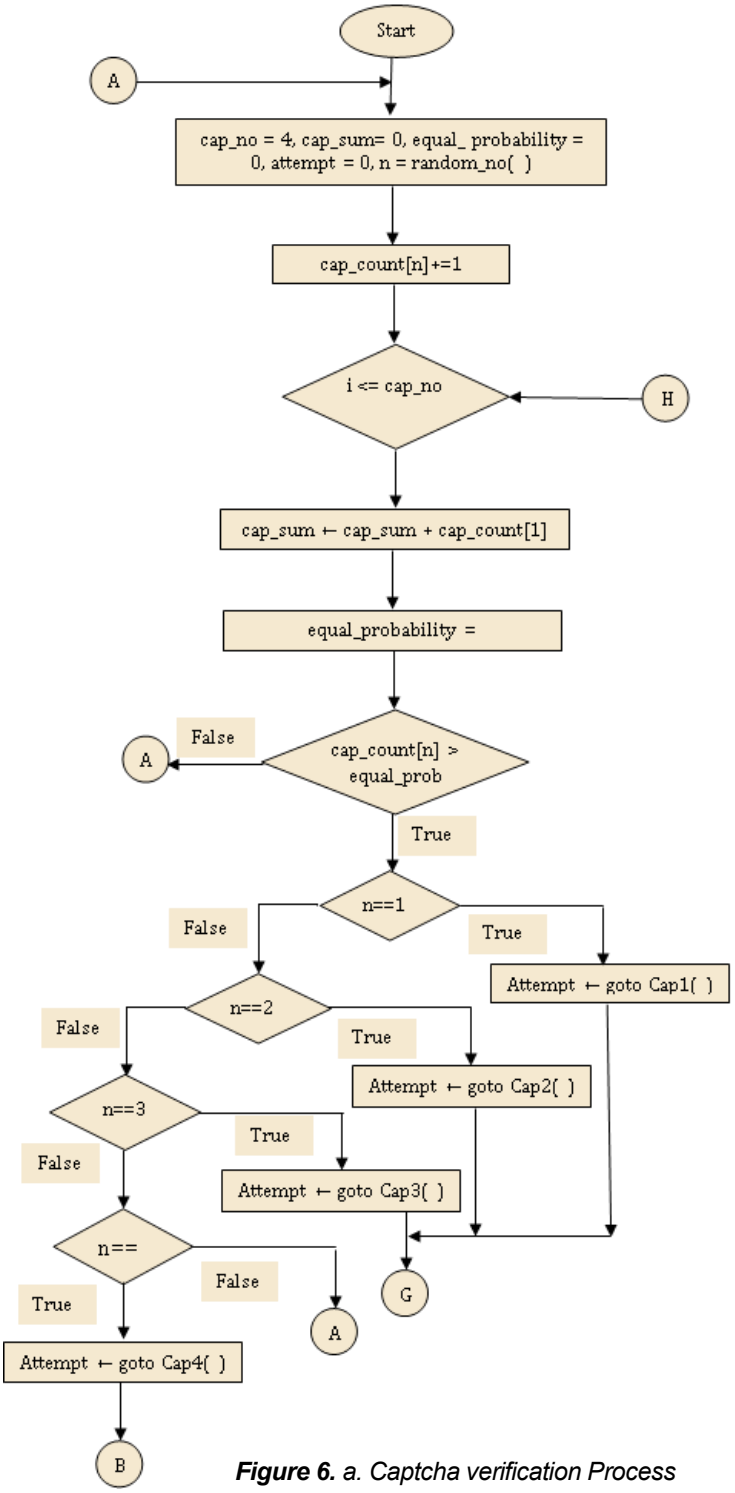

Figure 6. *a. Captcha verification Process*

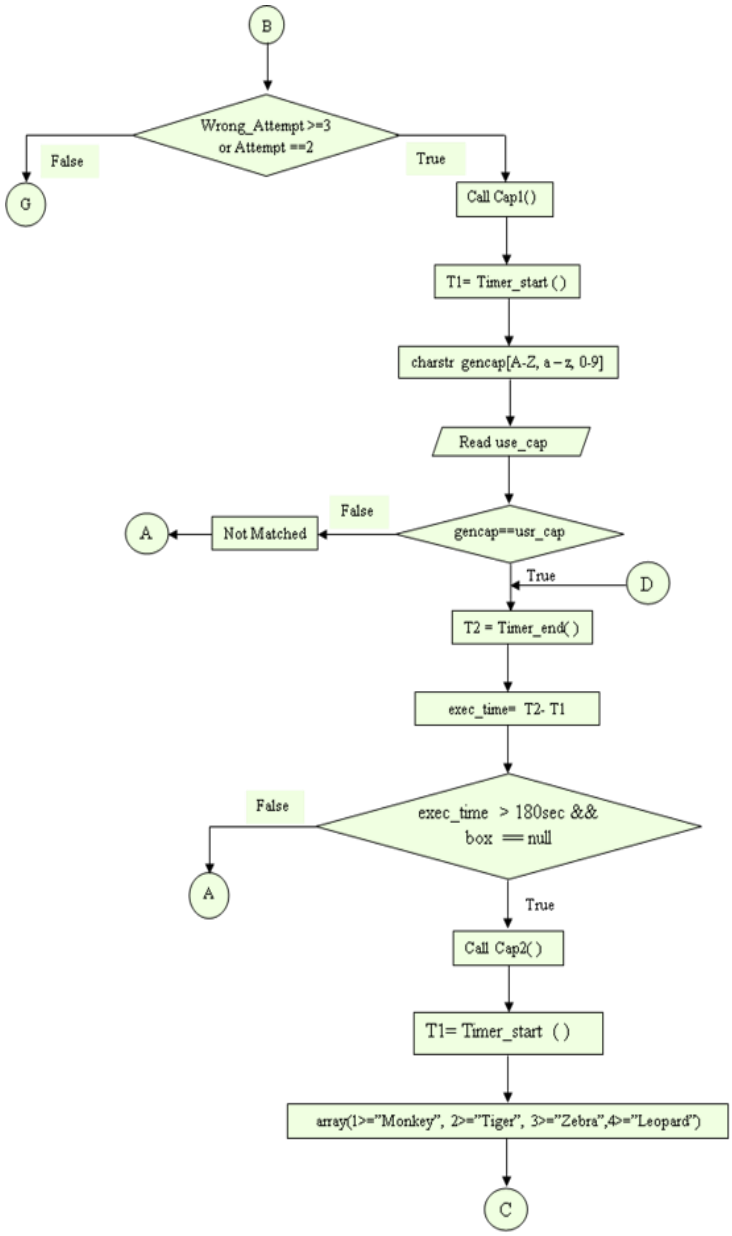

Figure 6. b*. Captcha verification – Functions Cap1 and Cap2*

Figure 6. d. Captcha verification– Cap4

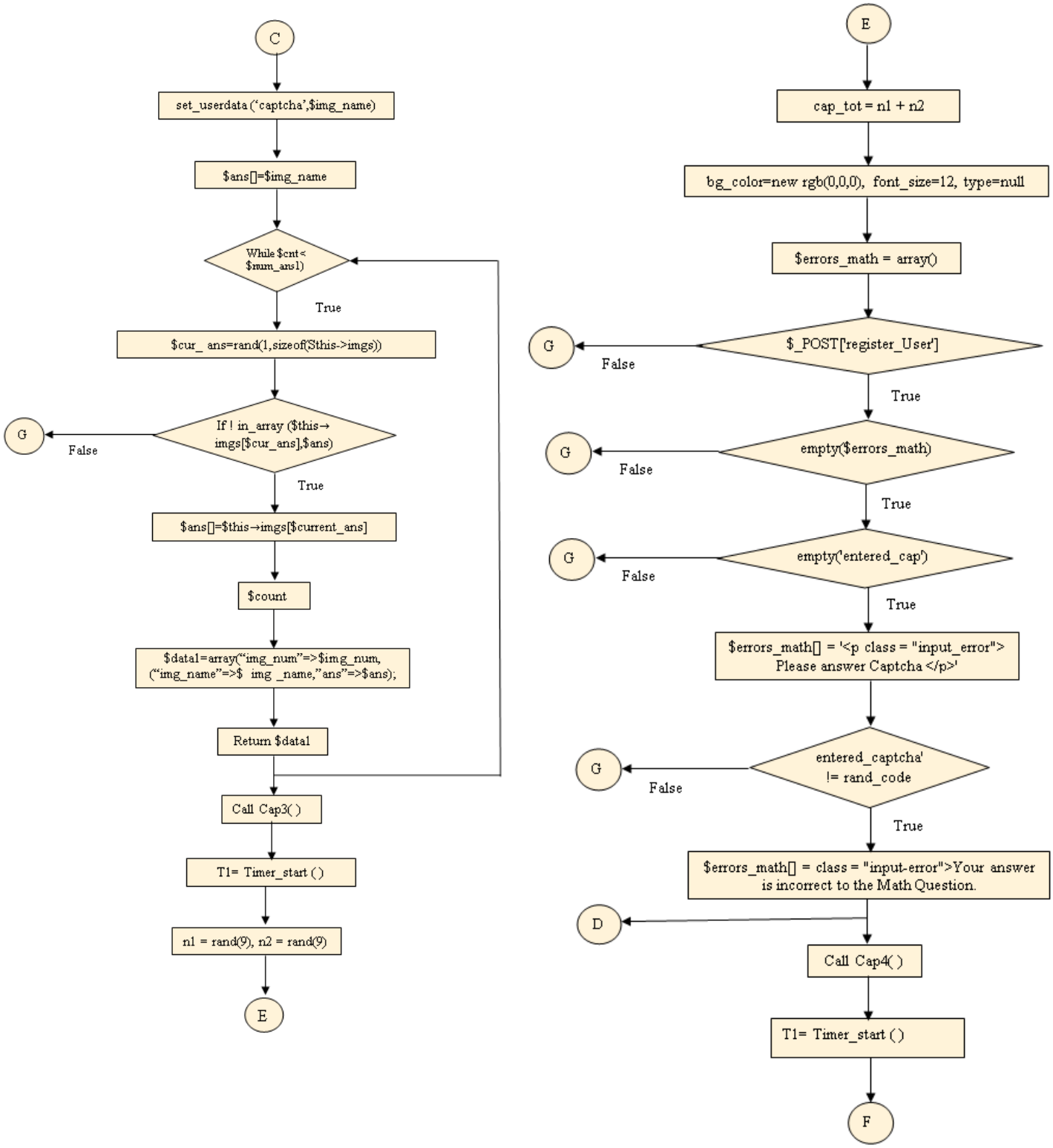

***Figure 6. c**. Captcha verification – Functions Cap2 and Cap3*

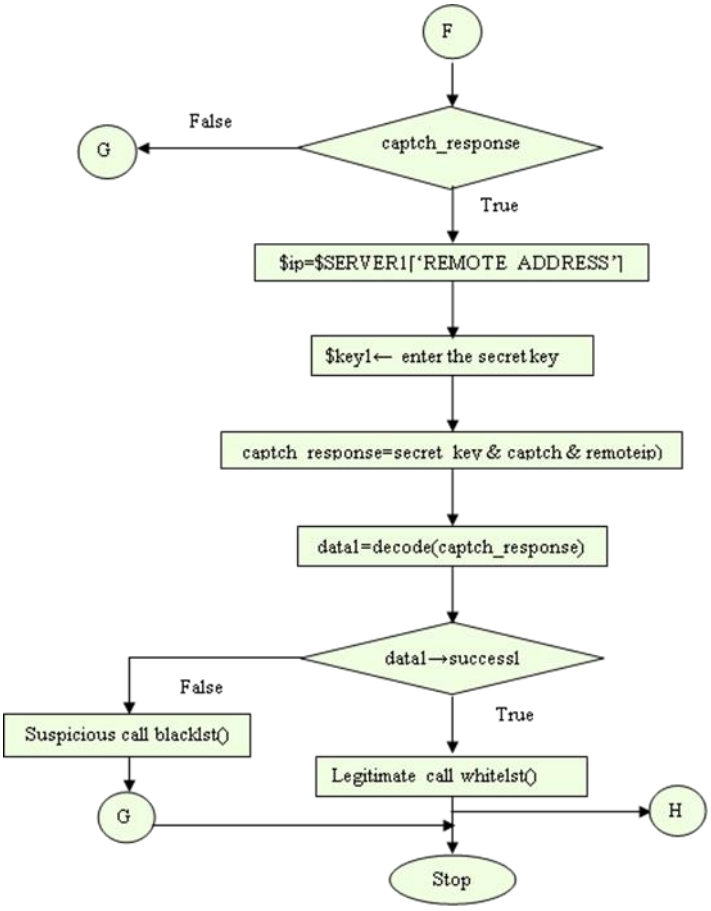

***Figure 6. e**. User Identification – Log Update*

3.2.2. Proposed Algorithm

Algorithm: CAPTCHA

1. Start
2. Declare cap_no, cap_count [cap_no], cap_sum, equal_ probability, int cap_sum, attempt, n, i
3. Initialize variables
   a. cap_no ← 4
   b. cap_sum← 0, equal_ probability ← 0
   c. attempt ← 0
   d. n ← random_no( )
   e. i←1
4. Increment cap_count[n] (for every random numbers generated by incoming users)
   a. while i <= cap_no do
   b. cap_sum ← cap_sum +cap_count[1]
   c. end while
5. User could appear only 3 times in CAPTCHA test
   a. equal_probability ← cap_sum/cap_no
6. if cap_count[n] > equal_prob
   a. goto Step 2

   else
      i. if n==1
         1. Attempt ← goto Cap1( )
      ii. else if n==2
         1. Attempt ← goto Cap2( )
      iii. else if n==3
         1. Attempt ← goto Cap3( )
      iv. else n==4
         1. Attempt ← goto Cap4( )
      v. end if
   b. end if
      i. if Attempt==1
         1. Increment Wrong_Attempt
      ii. else
      iii. goto the second verification process
      iv. end if
      v. if Wrong_Attempt >=3 or Attempt ==2
         1. end the process
      vi. else
      vii. goto Step 2
      viii. end if
7. Include CAPTCHA client and inform the client with the respective position
   a. string usr_cap;
      i. Over CAPTCHA Enter: ";
      ii. usr_cap;
   b. if checkcap (captch, usr_cap) is true
   c. CAPTCHA Matched
   d. else
   e. CAPTCHA Not Matched
   f. return 0;
8. Set the execution time for about 180 seconds
   a. T2 = Timer_end( )
   b. Convert millisecond to second execution_time= T2- T1
   c. if execution_ time > 180 seconds and box == null
      i. exit all the process
      ii. goto Step 3
      iii. else
      iv. if answer !=true
      v. goto Step 2
   else
      i. return(0)
      ii. end if
   end if

End CAPTCHA

Algorithm: Cap1

9. Generate Text Captcha
10. Set the timer condition T1= Timer_start ( )
    a. Bool check_cap (capt & text, string & user_ cap)
    b. return cap.compare (user_cap) == 0;
11. Compare user input
    a. String generatecaptch(int n)
       i. time_t t;
       ii. srand((unsigned)time(&t));
    b. char * charstr = "abcdefghijklmnopqrstuvwxyzABCDEFGHI JKLMNOPQRSTUVWXYZ0123456789";
    c. string cap = " ";
    d. while n-- do
    e. captch.push_back(chrs [rand ( ) % 62]);
    f. return cap;
    g. end while
    h. string cap= generateCap(9);
    i. write cap;

End Cap1

Generate CAPTCHA automatically to test client and return a valid CAPTCHA when there are two different strings. It is intended that it would include all the lowercase, uppercase

and the numbers ranging from 0-9. The CAPTCHA length is 62 and every time a random CAPTCHA should be generated from charstr [ ] array. The CAPTCHA with given string length of 62 characters will be created and the generated CAPTCHA will be compared with users input CAPTCHA.

Algorithm: cap2

12. Create image CAPTCHA Cap2( ). /*The images are chosen randomly. */
13. Repeat Step 8.
14. Verification Process:
    a. Var images← array(1>=”Monkey”,2>=”Tiger”,3>=”Zebra”,4>=”Leopard”)
    b. Function gen_cap (num1_answer)
    c. img_num=rand(1,Sizeof(this→imgs))
    d. img_name=this→ imgs (img_num)
    e. this→CI→session→set_userdata(‘captcha’,img_name)
    f. ans[]=img_name;
    g. Next, additional options num ans1 must be found cnt=0;
    h. While cnt < (num_ans1) do
    i. current_ ans=rand(1,sizeof(this->imgs))
        i. If !in_array(this→imgs[current_ans],ans)
        ii. ans[]=this→imgs[current_ans]
        iii. Increment count
        iv. End if
    j. End while
15. Shuffle answer set to prevent the first response from being correct answer
16. Build the data array
    a. data1=array(“img_num”=>img_num, (“img _name”=>img _name,”ans”=>ans);
    b. Return data1;
17. Repeat Step 12

End Cap2

Image verification process is done with random image creation and setting out proper response. Create array of images of animals and these array of images should be determined from that particular location.

Algorithm: Cap3

18. Create mathematical CAPTCHA Cap 3( ).
19. Repeat Step 8.
20. Initialize n1 and n2 with random numbers from 1 to 9
21. Add n1 and n2 and store it in cap_tot
22. Initialize the back ground colour, font size and type and the expression could appear as an image.
    rand_code= cap_tot
    a. font1 ← 'mmobuyit-captcha-fonts/Times new Roman.ttf'
    b. img ← imgcreatet_color(120, 30)
    c. black_color ← imgcolor_allocate(img, 0, 0, 0)
    d. color1 ← imgcolor_allocate(img, 0, 100, 90)
    e. white_color ← imgcolo_allocate(img,0, 26, 26)
    f. imgfill_edrectngle(img,0,0,399,99,white_color)
    g. img_ttftext (img, 20, 0, 20, 25, color1, font1, math1);
    h. header("Cont_type: img/png")
    i. imgpng(img)
23. errors_math = array()
    a. if 'register_User'
        i. if empty(errors_math) === true
            1. if empty('entered_cap') errors_math[] = Answer the Captcha Question
            2. Else if entered_captcha != rand_code errors_math[] = Your answer is incorrect to Math Question.
            3. End if
        ii. End if
    b. End if
24. Repeat Step 12

End Cap3

Algorithm: Cap4

25. Create Cap4 ().
    /*Here the system should provide I’m not a robot */
26. Repeat Step 8
27. Create class with the secret key
    a. form name=”Form1_Name” method=”post”action=”Dest1_action”
    b. class=”captch” data_sitekey=”SITEKEY_INSERT”
    c. If isset(captch_response)
    d. captch=captch_response
    e. ip=SERVER[‘REMOTE_ADDRESS’]
    f. key1← enter the secret key
    g. url=https://www.google.com/captcha/siteverify
    h. captch_response= get_contents(secretkey, response_captch, remote_ip);
    i. data1= decode(captch_response)
        i. If isset(data1→success1)&& data1→success1==true
            Print Enter legitimate user
        ii. Else
            Print Account is blocked
        iii. end ifs
    j. end if

Repeat Step 12.

End Cap4

Mitigation system of Distributed Denial of Service proposed architecture use Dynamic Captcha Testing with Equal Probability Algorithm. It is used for first verification process, to strengthen security in Client side network. Furthermore to overcome distributed denial of service attack in integration of Internet of Things and Cloud Environment. Firewall conducting two verification test such as Dynamic Captcha test and Puzzle test is used to identify whether incoming request is from legitimate user or malicious user. In first verification process use Dynamic Captcha Testing with Equal Probability Algorithm to recognize malicious user or legitimate user. This algorithm using simple Captcha test such as object pictures, distorted text, mathematical calculation and I m not a Robot is used for user authentication in client side network. Previous methodologies proposed for text Captcha to identify malicious user, limitations: the first is, author

providing permissions to enter captcha through more attempts. Hence malicious user may break the test through repeated attempts using programs or manual effort. Hence to overcome this proposed architecture introduces one unique feature; that is letting user to enter captcha within three attempts, within these three attempts user entering the correct Captcha is a legitimate user, otherwise malicious user/attacker. Suppose legitimate user answer captcha within these three attempts enter into second verification test. Malicious user didnot answer within three attempts, control exits from first verification test and no more attempts will be provided. The second limitation in previous methodologies is users given more time nearly an hour to enter captcha, as the malicious users may break the test using automation/ manual approach. Hence to overcome this proposed architecture uses unique feature. Letting user to enter Captcha within 3 minutes, assuming within this 3 minutes if user enter correct Captcha the user is legitimate, otherwise malicious user. If legitimate user some time didn't respond within specified time interval (3 minutes) also if Captcha box is null even if it is legitimate user, control is let to exit from first verification test. As both, legitimate or malicious user did not answer within 3 minutes control leaves first verification test. Assuming legitimate user should answer Captcha within allocated minutes enter into second verification test. Third unique feature in proposed architecture is dynamic test. Assume 1000 user sending incoming request at a time, author conducting i m not a robot test for all 1000 user, attacker may easily identify Captcha and may break the test. Hence proposed architecture presents dynamic Captcha test with equal probability. Here four types of Captcha test such as object pictures, distorted text, mathematical calculation and I m not a Robot test is equally allocated for 1000 users, and also randomly four test is allocated for 1000 users. Based on this, malicious user may not recognize which test is meant for first verification process. Along with these novel features proposed architecture is found superior then exiting architectures.

4 EXPERIMENT AND RESULT ANALYSIS

An initial test bed experiment was conducted with the proposed system, MSOFDDoS, which demonstrates that the test bed works well with regard to firewall success in filtering received packets by frame algorithm and the success of the method used to capture received packets in efficient performance of the task. The first is the protected server, the second is the client's side and the third is the firewall with two internal and external interfaces. For the implementation, a Linux platform is used.

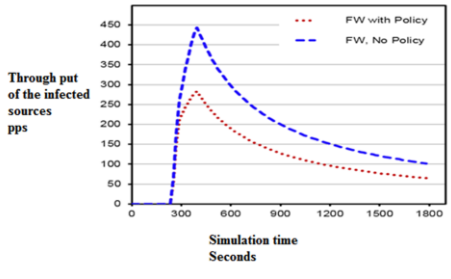

Figure 7 *Average throughput for HTTP packets that are initiated from the infected devices in packers per seconds(pps).*

Infected devices Throughput

The average throughput for HTTP output from infected devices represents the performance of the protection program being enforced, with and without a firewall rule, which is being targeted. Figure 7 shows the use of a generic Firewall rule to address such malicious requests does not make any difference. The peak average performance increases sharply until the synchronization between the various components for HTTP applications reaches 445 packets per second (pps) without a policy and 280 pps with Firewall Policy seven minutes after installation. This then decreased respectively to a rate of 100 to 60pp. This reduction demonstrates the efficiency of the firewall and the proposed policy firewall, in particular malicious requests attempting to gain access to the secured server.

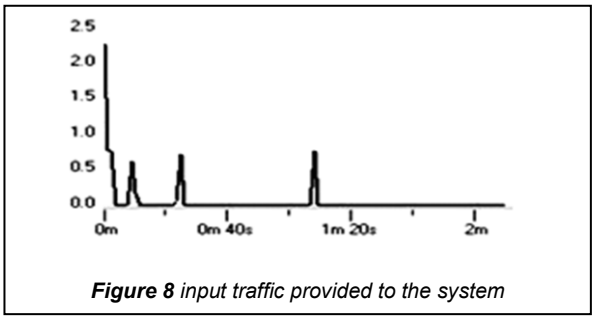

Figure 8 *input traffic provided to the system*

Figure 8 indicates the speed of response, examination of packets, and their arrival time are clearly. Here earlier packets have been arrived faster than the other packets and that was almost 4 times greater than other iterations.

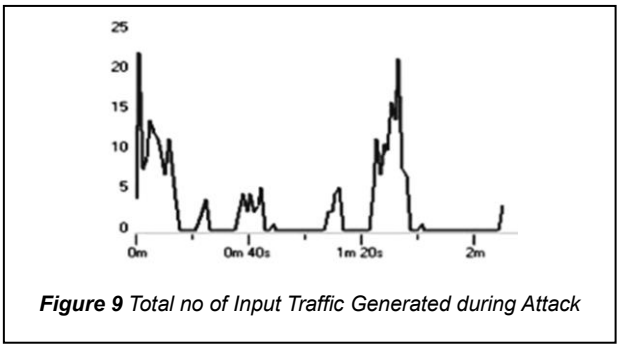

Figure 9 *Total no of Input Traffic Generated during Attack*

As the statistics was collected from network traffic as shown in Figure 9 though this measure of efficiency is related to attack that was assumed to take place in the current scenario. After the analysis of intruders, proposed mechanism have been implemented and result is being generated as it is depicted in Figure 10. This shows the frequency of attacks undertaken by clients and it is found to be lesser nearing 10%, based on input provided.

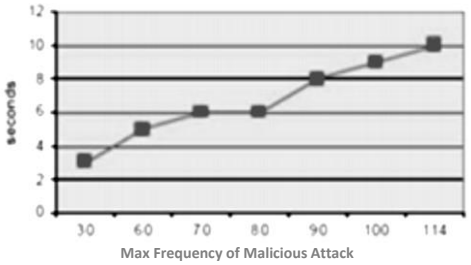

Figure 10 *Maximum frequency of attack detected*

As results confirm that proposed mechanism and flow that is introduced could withstand for a greater number of attacks and could perform well for legitimate users. While resisting malicious programs as well as users by including multiple CAPTCHA, authorized users may not suffer delay in response, thereby improving efficiency of cloud services.

5 CONCLUSION

A DoS has been clarified by its definition, process, and requirements to be met in order to defend the cloud against such attacks by any proposed solution. Some of the latest DDoS attacks approaches was discussed and assessed. Dynamic Captcha Tests with Equal Probability Algorithm is proposed in this paper for the first test to check the Authentication of client side networks for the purpose of addressing distributed denial of services attacks in Internet of Things and cloud environments. This paper also explains the types of Captcha approaches, processes and pertinence for proposed system. Therefore, a comparison has been made between specific existing solutions to protect the scalability, check incoming requests and proactively protect customer networks. It can be used to combat DDoS attacks by inspecting the origin of requests (legitimate or malicious) to confirm its validity using Fisrt Captcha test and second validation puzzle test, followed by monitoring of other packages via an Intrusion Prevention Program and a Reversing Proxy server, to examine malware components that can be contained inside them. Two more key lists have been added to the lists of questionable and malicious firewalls, to boost reaction to DDoS-assault. To prove its validity and efficiency, proposed model was evaluated.

REFERENCES

[1] Gubbi J, Buyya R, Marusic S, Palaniswami M, "Internet of Things (IoT): A Vision, Architectural Elements and Future Directions", Future Generation Computer Systems, Vol. 29, pp.1645–1660, ELSEVIER, 2013, DOI: /10.1016/j.future.2013.01.010.

[2] NIST," Global City Teams Challenges- Smart America Round Two" http://www.nist.gov/cps/upload/20140723-Smart America Global City-Teams -Challenge-introduction-v1-6p.pdf.

[3] Mell, P., Grance, T., The NIST definition of cloud computing. National Institute of Standards and Technology 53 (6), 50, 2009.

[4] S. Sivakumar, V. Anuratha, S. Gunasekaran, "Survey on Integration of Cloud Computing and Internet of Things Using Application Perspective," International Journal of Emerging Research in Management &Technology, ISSN: 2278-9359 (Volume-6, Issue-4). April 2017.

[5] M. Waterhouse, "Rutgers University's computer network under DDoS attack", http://abc7ny.com/technology/rutgerscomputer-network-undera ttack, website-internet-access-down-on campus/1006255, 2015.

[6] Y. Xia. Cloud control system. IEEE/CAA of Journal of Automatica Sinica, 2015, 2(2): 134 – 142.

[7] Z. Xu, Q. Zhu. Secure and resilient control design for cloud enabled networked control systems. Proceedings of the 1st ACM Workshop on Cyber-Physical Systems-Security and/or PrivaCy, Denver: ACM, 2015: 31 – 42.

[8] J. Wan, S. Tang, H. Yan, et al. Cloud robotics: current status and open issues. IEEE Access, 2016, 4: 2797 – 2807.

[9] B. Kehoe, S. Patil, P. Abbeel, et al. A survey of research on cloud robotics and automation. IEEE Transaction on Automation Science and Engineering, 2015, 12(2): 398 – 409.

[10] C. Johnston S. Thielman. "Major cyber attack disrupts internet service across Europe and US". In: The Guardian (Oct. 21, 2016). URL: https://www.theguardian.com/technology/2016/oct/21/ddos-attack-dyn-internet-denial-service.

[11] Robert Bronte, Hossain Shahriar, and Hisham M. Haddad. "Mitigating Distributed Denial of Service Attacks at the Application Layer". In: Proceedings of the Symposium on Applied Computing. SAC '17. Marrakech, Morocco: ACM, 2017, pp. 693–696. ISBN: 978-1-4503-4486-9. DOI: 10.1145/3019612.3019919.

[12] N Muraleedharan and B Janet. "Behaviour analysis of HTTP based slow denial of service attack". In: Wireless Communications, Signal Processing and Networking (WiSPNET), 2017 International Conference on. IEEE. 2017, pp. 1851–1856.

[13] Seyed Milad Helalat. "An Investigation of the Impact of the Slow HTTP DOS and DDOS attacks on the Cloud environment". MA thesis. Blekinge Institute of Technology, 2017, p. 74.

[14] Nikhil Tripathi and Neminath Hubballi. "Slow rate denial of service attacks against HTTP/2 and detection". In: Computers & Security 72 (2018), pp. 255–272. ISSN: 0167-4048. DOI: https://doi.org/10.1016/j.cose.2017.09.009.

[15] Dan C Marinescu. Cloud computing: theory and practice. Morgan Kaufmann, 2017.

[16] S. S. Chopade, K. U. Pandey, and D. S. Bhade, "Securing cloud servers against flooding based DDOS attacks," Proc. - 2013 Int. Conf. Commun. Syst. Netw. Technol. CSNT 2013, pp. 524–528, 2013.

[17] J. B. D. Joshi, H. Takabi, and S. T. Zargar, "DCDIDP: A distributed, collaborative, and data-driven intrusion detection and prevention framework for cloud computing environments," in 7th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2011, pp. 332–341.

[18] W. Dou, Q. Chen, and J. Chen, "A confidence-based filtering method for DDoS attack defense in cloud environment," Futur. Gener. Comput. Syst., vol. 29, no. 7, pp. 1838–1850, 2013.

[19] M. Malekzadeh, A. Ghani, and S. Subramaniam, "A new security model to prevent denial-of-service attacks and violation of availability in wireless networks," Int. J. Commun. Syst., vol. 25, no. 7, pp. 903–925, Jul. 2012.

[20] R. Anandhi and V. N. Raj, "Prevention Of DDoS Attacks On Distributed Cloud Servers By Port Lock Mechanism," ARPN J. Eng. Appl. Sci., vol. 11, no. 5, pp. 3013–3019, 2016.

[21] M. Rahman, W. M. Cheung, "A Novel Cloud Computing Security Model to Detect and Prevent DoS and DDoS Attack," J. Adv. Comput. Sci. Appl., vol. 5, no. 6, pp. 119–122, 2014.

[22] C. Borean, R. Giannantonio, M. Mamei, D. Mana, A. Sassi, "Internet and Distributed Computing Systems," vol. 9258, pp. 143–154, 2015.

[23] Mayur Patel, NimitModi,"Authentication Using Graphical Password", International Journal of Computational Engineering Research (IJCER) ISSN (e): 2250-3005 Vol, 04 Issue, 11 November 2014.

[24] Rahman UR., Tomar D.S., Das S., "Dynamic Image Based CAPTCHA", International Conference on Communication Systems and Network Technologies (CSNT)., pp. 90-94, May 2012.

[25] Huang V.S., Huang R., Ming Chiang, "A DDoS Mitigation System with Multi-Stage Detection and Text-Based Turing Testing in Cloud Computing", 27th International Conference on

Advanced Information Networking and Applications Workshops (WAINA)., pp. 655-662, March 2013.
[26] Von Ahn, L., Blum, M., Nicholas, J.H., Langford, J.," CAPTCHA: Using Hard AI Problems For Security", In Proceedings of Eurocrypt, pp.294-311, 2003.
[27] Shahreza, M., Shahreza, S., "Preventing Mobile Software Cracking Software", IEEE, Innovations in Information Technology, Dubai, 2006, pp. 1-5.
[28] Microsoft Hotmail, http://www.hotmail.com [06/10/2008]
[29] Pan Lei, Zhou Yan, "Developing an Empirical Algorithm for Protecting Text-based CAPTCHAs against Segmentation Attacks", 12th IEEE International Conference on Trust Security and Privacy in Computing and Communications (TrustCom), pp. 636-643, July 2013.
[30] Thomas V.A, Kaur K., "Cursor CAPTCHA– Implementing CAPTCHA Using Mouse Cursor",Tenth International Conference on Wireless and Optical Communications Networks (WOCN)., pp. 1 - 5, July 2013.
[31] Tamang T., Bhattarakosol P. "Uncover impact factors of text-based CAPTCHA identification", 7th International Conference on Computing and Convergence Technology (ICCCT). pp. 556 – 560 Dec. 2012.