Periodicals of Engineering and Natural Sciences ISSN 2303-4521
Vol.6, No.1, June 2018, pp. 161~170
Available online at: http://pen.ius.edu.ba

Analysis of the EDoS Attack impact on Elastic Cloud Services Using Finite Queuing Model

Suneetha Bulla[1], B. Basaveswara Rao[2], K. Gangadhara Rao[3], K. Chandan[4]

[1]Department of CSE, Acharya Nagarjuna University, suneethabulla@gmail.com
[2] University Computer Centre, Acharya Nagarjuna University, bobbabrao@yahoo.co.in
3 Department of CSE, Acharya Nagarjuna University, kancherla123@gmail.com
[4] Department of Statistics, Acharya Nagarjuna University, kotagirichandan@gmail.com

Article Info

Article history:

Received Aug 12th, 2017
Revised Mar 12th, 2018
Accepted Jun 16, 2018

Keyword:

EDoS
Finite Queue
legitimate
malicious

ABSTRACT

This paper proposes a logical model to examine the effect of the EDoS attack in cloud environment using finite queuing model and enhanced with experimental model. Due to this sophisticated attacks the computing resources are busy and buffer capacity of the cloud gets exhausted by both the legitimate and malicious user requests, because of this both types of requests could not get the service. The legitimate customers are unable to get service of web application. In this backdrop this paper investigates and evaluates the vendor loss factor from the cost factor of view since the legitimate client requests are denied service. The objective of this analysis is twofold i) to identify the dynamics of the EDoS attacks with different attack rates and to measure the various performance metrics (total number of busy virtual machines, utilization of the cloud resources, request response time, request loss probability, and throughput). ii) The cost function is defined and evaluated based on these performance metrics. Finally compared analytical and experimental results are presented and conclusions are drawn.

Corresponding Author:

B. Basaveswara Rao,
University Computer Centre,
Acharya Nagarjuna University,
bobbabrao@yahoo.co.in

1. Introduction

Cloud computing (CC) is sharing virtual resources or services like computing resources, storage databases, web applications and other cloud services through the internet using pay-as-you-go basis. Gartner Identified CC is one of the top ten technologies of the IT industry and he predicted cloud revenue to grow 21.4 percent in 2018 [1]. The global adoption of this technology reduces the cost factor of users and organizations. Organizations are migrating businesses into the cloud, so that they can rent the cloud services for use on a subscription instead of building their own infrastructures using pay-as-you-go basis.

There are a couple of sorts of strikes which damage the computing assets and associations of customer cloud environment and it leads to compromise SLA. In light of SLA, cloud assets are given to client in confined or limitless mode [2]. The preferred standpoint utilization and the taking care of intensity are charged to the customer. The vendors cloud resource cost can be incurred by the attacks.

Distributed Denial of Service (DDoS) attack is one of the major assaults in the CC and the fundamental objective of the DDoS attack is leads the resource unavailability, reducing the v by damaging the virtual servers [3]. Distributed Denial of Service (DDoS) assaults target goals, energized applications or system structures by connecting all open transmission confine and aggravating access for genuine clients and partners.

According to the official National Institute of Standards and Technology (NIST) definition, Cloud figuring is 'a model for enabling unavoidable, profitable, ondemand mastermind access to a typical pool of configurable enrolling resources - for example, frameworks, servers, accumulating, applications and organizations - that

can be immediately provisioned and released with unimportant organization effort or expert center interaction'[22].The NIST defines rapid elasticity is the one of essential characteristics of CC [4]. This scalability mechanism shows solution to the DDoS attacks by handling the web application traffic using scaling up and down the cloud resources, at the same time it introduced new sort exhibit of assault named as Economical Denial of Sustainability (EDoS). The EDoS is defined as an attack that affects the vendors bill by sending the http proxy that act as legitimate, utilize the elasticity property of the cloud resources, parallel to traditional DDoS attacks [5]. When the elasticity can be finite and it can affect by the EDoS attack and leads to request loss probability of the legitimate user. So it is necessary to research of this attack of this scenario to achieve the current information.

Many of researchers used queuing method to identify the effect of the EDoS assault on CC, but they are restrict to infinite queuing model and taken an assumption of user request loss probability is zero. In real scenario the cloud resources are finite and they are affected by the malicious users. This paper fulfilled this gap and proposed an analytical model to achieve the maximum elasticity of the typical cloud data center and derived the effect of the EDoS assault in terms of request loss probability, the impacresponse time, cloud use, cost of the advantages and throughput of the of the EDoS assault on legitimate users.

Whatever remains of the paper is composed as pursues. The writing overview displays in area 2. Segment 3 portrays our expository model to catch the conduct of an EDoS assault in the flexible cloud benefit facilitated on a cloud. The segment determines equations that can be utilized in foreseeing key execution estimates that can be utilized in accomplishing appropriate flexibility in wording and loss probability of the traffic. Section 4 discussed about the experimental test-bed setup to support the analytical model. Section 5 analytical results are compared with experimental results and discussed. Finally, Section 6 completes the paper and formats future work.

1. Related work

Khaled Salah et.al [6], in this article author proposed sensible model nearby a total calculation for choosing extra key execution conditions and measures. In this paper, another region has been joined appearing and investigating the versatility of the LB that can be, at whatever point ignored, an important execution bottleneck for versatile organizations. Another domain was consolidated avowing our deliberate model and conditions. The underwriting was composed utilizing estimations of an exploratory demonstrating ground sent on the AWS cloud. Just more fundamentally, this paper solidifies numerical consequences of veritable functional conditions of cloud versatile organizations that wire web advantage, Netflix video spouting, and the AWS cloud. The area on numerical outcomes joins new figures and essential exchange and understandings on cloud resource estimation and cutoff building edges related to achieving genuine adaptability for cloud organizations.

F. Al-Haidari, M. Sqalli, et al. [7], In this paper, showed that an investigative model to look at the effect of EDoS assaults on a particular class cloud benefits in which there is just a singular kind of use advantage gave in the datacenter. The model thinks about various execution estimations. These estimations join end-to-end reaction time, usage of getting ready assets being eaten up, and the accomplished expense happening in light of the assault. Such model is valuable to show the effect of an EDoS assault on both execution and cost of the spread preparing associations. Suneetha Bulla, B Basaveswara Rao et al. [8], this authors are enhanced [7] analytical model by using the experimental model. In this paper proposed experimental model and explained how to deploy experimental model on the AWS cloud to study the performance and cost impact of the EDoS attacks.

Khaled Salah [9], proposed a logical method to achieve the elasticity of the cloud using cluster jobs and validated using simulation. The main prediction on this paper is elasticity and he gave numerical example to illustrate and demonstration of the queuing model.

Gian-Luca Dei Rossi, Mauro Iacono, and Andrea Marin [10] proposed a Markovian model to consider the effect of eDoS strikes to cloud infrastructures. This analysis is depending up on the assessment of the mean time to absorption and on the expected cumulated rewards in a CTMC describing the attacker strategy and the cloud state. This model gave numerically stable methods to compute (or approximate for long-lasting attacks) the performance indices that allow us to evaluate the impact of an attack.

Shi et al. [11] have made convincing centrality sparing methods in the cloud datacenter by consistently assigning assets dependent on usage examination and guess. The rule measure plot that has been utilized in their work was a M/M/1 lining model that gets the cloud-based web advantage. Fundamentally, Calheiros et al. [12] have proposed a versatile provisioning system for cloud-constructed preferences for go in light of cloud-based applications that meet QoS targets subject to covering system structure appear and remaining employment waiting be done data. They demonstrate each virtualized application point of reference as a M/M/1/k lining model, where k proposes an obliged line of length k.

2. Analytical model

This paper aiming to analyze or evaluate the impact of the EDoS attacks on finite elastic cloud hosted web services. Figure 1 shows the architecture of elastic cloud datacenter, this datacenter contain software as a service type web application [7] and providing single service. This architecture contains three phases those are Load Balancer, elastic group of Virtual Machines (VM) and Database Server. Legitimate users are utilize this services based on the SLA, but attackers are targets this elasticity nature of the cloud to unavailability of service to the customers and increase the cost of the cloud adopters.

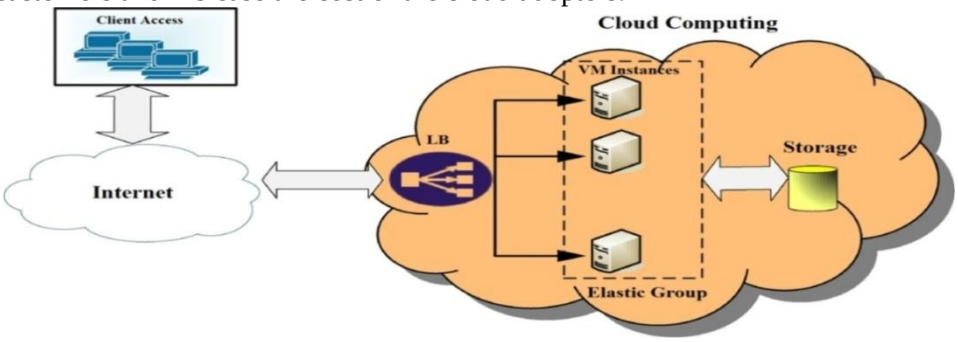

Fig 1: Elastic cloud services of AWS

Figure 2 illustrates about queuing model which is a representative of cloud hosted single web service architecture shown in the figure 1. In this model consider that the arrival rate follows passion distribution and assumed service time follows exponential distribution for all instances of ECS including computation as well as latency of DB server and band width of the network. Buffer size of the first queue is K-1 and arrivals are getting serviced sequentially following stages: (1) load balancer, (2) elastic computing resources and (3) database server.

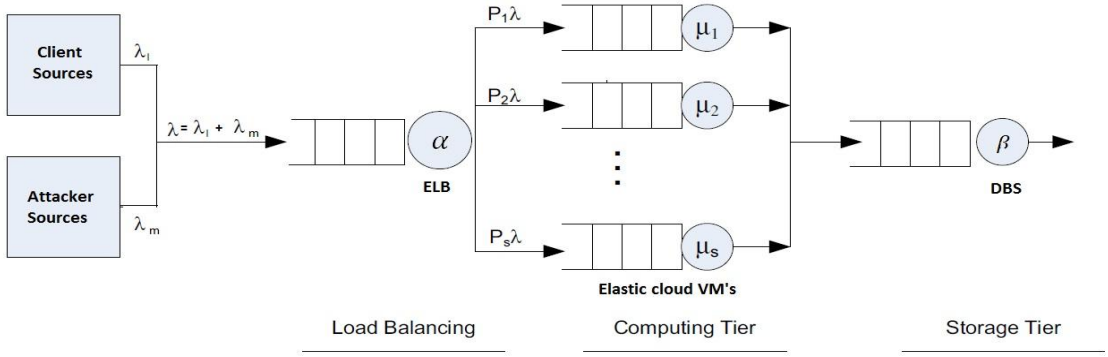

Fig 2: Queuing Model for the web application Hosted on Elastic Cloud services

Where λ_l is the mean arrival rate of legitimate users
λ_m is the arrival rate of the attackers
α is the service rate of the ELB
μ is the service rate or capacity of the virtual machines
β is the service rate of the database server
S is the total number of servers running in the cloud
K capacity of the queuing system per second
γ is the throughput rate of the queuing system

$$\lambda = \lambda_l + \lambda_m$$

The above queuing model follows M/G/1/K settling structure with Poisson areas λ, for the most part scattered association times and most extreme of the framework is K. z(x) implies the figure of the covering framework which is the association times of the PDF of everything considered passed on self-self-assured variable X, Ek is the likelihood of having k entries of amidst a specific association time and is conferred as

$$E_k = \int_0^\infty \frac{((\lambda l + \lambda m)x)^k}{k!} e^{-\lambda x} z(x)dx \quad (1)$$

Actually z(x) is the sum of the three service stages those are ELB, VM's and database , those service times has an exponential distributions. First and third stages mean and PDF's are respectively 1/α, 1/μ and $f(t) = \alpha e^{-\alpha t}$ and $g(t) = \beta e^{-\beta t}$. The second stage follows parallel computing with available instances, then μis the static service time for these S parallel severs, here μ can be expressed $\mu_i = \mu$. Third stage mean and PDF's are 1/ μ and $f_{B(t)=S\mu(1-e^{-\mu t})^{S-1}e^{-\mu t}}$.

Then, z(x) is derived from these three PDF's by integration. Convolution is direct administrator and figure first convolution of first and third PDF's and the resulting function is h (t) and after that calculate the convolution of h(t) and f_B**(t).** In [13] Takagi has been discussed the convolution of two density functions of two random service times with two means , then in our queuing system compute convolution of f(t) and g(t) with means of 1/β and 1/α can be expressed as

$$h(t) = \begin{cases} \frac{\beta\alpha}{\alpha-\beta}\left(e^{-\beta t} - e^{-\alpha t}\right) & \alpha \neq \beta \\ \alpha^2 t e^{-\alpha t} & \alpha = \beta \end{cases} \quad (2)$$

From the expression (2) there are two condition, first condition indicates there is loss probability of arrivals and second one is there is no loss probability. Let us consider first case where $\alpha \neq \beta$. z(x) is the convolution of h(t) and $f_B(t)$.

$$z(x) = \frac{S\beta\mu\alpha}{\alpha-\beta}\left(\int_0^x (1-e^{-\mu t})^{S-1} e^{-\mu t}\, e^{-\beta(x-t)}dt - \int_0^x (1-e^{-\mu t})^{S-1} e^{-\mu t} e^{-\alpha(x-t)}dt\right) \quad (3)$$

Solve the equation (3) by using integration by parts.

From equation (1) $E_k = \frac{(\lambda l + \lambda m)^k}{k!}\int_0^\infty x^k\, e^{-(\lambda l + \lambda m)x} z(x)dx$ and by substitution and solve the equation using integration by parts, then it gets E_k follows

$$E_k = \sum_{i=1}^{S} \frac{n\alpha\mu\beta C_{i,N} (\lambda l + \lambda m)^k ((\lambda l + \lambda m)+i\mu)^{-k-1}}{(\alpha-\beta)(\beta-i\mu)} - \sum_{i=1}^{S} \frac{n\beta\alpha(S-1)!\,(\lambda l + \lambda m)^k \mu^S ((\lambda l + \lambda m)+\beta)^{-k-1}}{(\alpha-\beta)\prod_{i=1}^{N}(\beta-i\mu)}$$
$$-\sum_{i=1}^{S} \frac{n\alpha\mu\beta C_{i,S} (\lambda l + \lambda m)^k ((\lambda l + \lambda m)+i\mu)^{-k-1}}{(\mu-r)(\mu-i\beta)} + \sum_{i=1}^{N} \frac{n\beta\alpha(S-1)!\mu^{N-1}\lambda^k\mu^N (\lambda+\alpha)^{-k-1}}{(\alpha-\beta)\prod_{i=1}^{S}(\alpha-i\mu)} \quad (4)$$

To summarize:

$$E_k = \left\{ \begin{matrix} \sum_{i=1}^{S} \frac{n\beta\alpha\mu C_{1,S}\lambda^k(\lambda+i\mu)^{-k-1}}{(\alpha-\beta)(\beta-i\mu)} - \sum_{i=1}^{S} \frac{n\beta\alpha(S-1)!\lambda^k\mu^S(\lambda+\beta)^{-k-1}}{(\alpha-\beta)\prod_{i=1}^{S}(\beta-i\mu)} \\ -\sum_{i=1}^{S} \frac{S\beta\alpha\mu C_{1,S}\lambda^k(\lambda+i\mu)^{-k-1}}{(\alpha-\beta)(\alpha-i\mu)} + \sum_{i=1}^{S} \frac{S\beta\alpha(S-1)!\mu^S(\lambda+\alpha)^{-k-1}}{(\alpha-\beta)\prod_{i=1}^{S}(\beta-i\mu)} \end{matrix} \quad \alpha \neq \beta \right\} \quad (6)$$

E_k is used figure the resolute state probabilities of the queuing system from ELB to database [19]. This presentation is used Imbedded Markov Chain To solve the steady state transition probabilities with following initial conditions, those are system state n_i denotes the type of processing takes place by ELB, Computing instances and database server. q_{jk} are the transition probabilities of the Imbedded Markov Chain at parity can be found independently using Ek for two cases j=0 and 1≤j≤K-1.

$$q_{0k} = \begin{cases} E_k & 0 \leq k \leq K-2 \\ & j = 0 \\ \sum_{s=k-1}^{\infty} E_s & k = K-1 \end{cases} \quad (7)$$

$$q_{jk} = \begin{cases} E_{k-j+1} & j-1 \le k \le K-2 \\ & 1 \le j \le K-1 \\ \sum_{s=k-1}^{\infty} E_s & k = K-1 \end{cases} \quad (8)$$

From the equation (7) and (8) we get persisting state probabilities {r_k :0≤k≤K-1} at the objective can be figured by clarifying K-1 counterbalance conditions and moreover with institutionalization condition as seeks after

$$r_k = \sum_{j=0}^{K-1} r_j \, q_{jk} \qquad 0 \le k \le K-1$$

$$\sum_{k=0}^{K-1} r_k = 1 \quad \text{(Normalization Condition)} \quad (9)$$

Substituting equations (7) and (8) into (9), then it get

$$r_k = r_0 E_k + \sum_{j=1}^{K+1} r_j \, E_{k-j+1} \qquad 0 \le k \le K-2 \qquad \text{and} \qquad \sum_{k=0}^{K-1} r_k = 1$$

From the equation (9) K-2 equations and along with normalization condition are used to unravel the arrangement of conditions to accomplish the unfaltering state probabilities r_k/r_0 and illuminate normalized variables using

$$\frac{r_{k+1}}{r_0} = \frac{1}{\alpha_0}\left[\frac{r_k}{r_0} - \sum_{j=1}^{k} \frac{r_j}{r_0} E_{k-j+1} - E_k\right] \qquad 0 \le k \le K-2 \quad (10)$$

The above equation is used solved recursively and successively determine $\{r_1/r_0, r_2/r_0, r_3/r_0, \ldots\ldots r_{k-1}/r_0,\}$. Subsequently, r_0 can be comprehended utilizing the standardization condition as pursues

$$r_0 = \frac{1}{\sum_{k=0}^{K-1} \frac{r_k}{r_0}} = \frac{1}{1+\sum_{k=1}^{K-1} \frac{r_k}{r_0}} \quad (11)$$

In this queuing system r_0 is used to get the particular state probabilities{r_k :0≤k≤K-1}.Assume P_k is the probability of k occupations presented in the covering structure at a self-confident time, where k = 400, 500, 600,…. , K. Using P_{loss} as the agreement probability that an arrival is lost the organization from merchant in light of the way that the line is full, that is in state K where P_{loss} and P_k are follows:

$$p_k = (1 - P_{loss}) r_k \qquad 0 \le k \le K-1 \quad (12)$$

$\bar{X}$ is sum of the mean service times.

$$\bar{X} = {}^{1}/_{\alpha} + E[B] + {}^{1}/_{\beta} \quad (13)$$

Where E[B] is the mean administration time organize. In the elastic cloud parallel computing resources are running in the particular time and providing single service. Tolerating that all the figuring cases have a comparative enrolling power restrain μi = μ and overhead made my submitting events the cloud organization to be 55.4s for provisioning one VM instances [33].

From equation (8) P_{loss} can be expressed as

$$P_{loss} = p_K = 1 - \frac{1-p_0}{\rho} = \frac{p_0+\rho-1}{\rho} \quad (14)$$

Where $\rho = \lambda \bar{X}$

Utilizing the estimations of ¬ rk got from Equation (10) and the consequences of (15) and (17), the harmony state dissemination {Pk:0≤k≤K-1 } can be communicated as

$$P_k = \frac{r_k}{r_0+\rho} \qquad 0 \le k \le K-1 \quad (16)$$

From Equation (16), $P_0 = \frac{r_0}{r_0+\rho}$ (17)

The throughput of departure γ also be called as the effective arrival rate λ' (or) λ (1-P_{loss}). Therefore

$$\gamma = \lambda (1-P_{loss}) \quad (15)$$

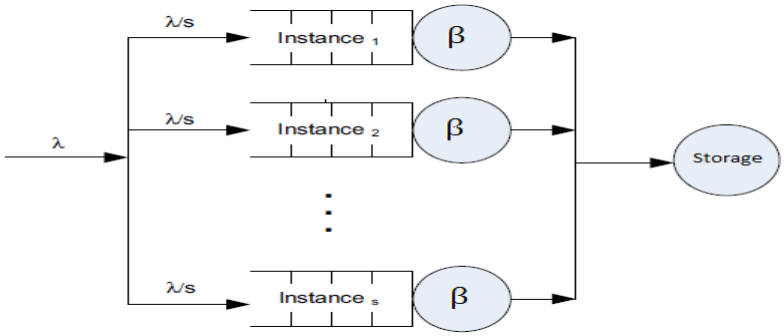

Fig 3: Queuing model of the computing elastic cloud

Figure 3 shows the open finite queuing model of the computing elastic cloud. In that the load balancer is handle the traffic and distribute this traffic to pool of available instances. Each instances arrival rate is $\lambda_i = \lambda/S$. where λ is the landing rate and S is the aggregate number of cases in the VPC. The mean computing utilization U and utilization effected by the attacker U_m are expressed

$$U = \frac{\lambda l + \lambda m}{s\beta} \text{ and } U_m = \frac{\lambda m}{s\beta} \quad (18)$$

Assuming that auto-scaling of the elastic cloud are configured with there is no delay to adopt new virtual machines into the elastic group [14]. Compare M/M/1 queuing model to derived equation of our queuing model with service rate $s\beta$, these two methods have a similar registering usage. The normal reaction time of a request to take service R and average response time effected by the victim R_m will be:

$$R = \frac{S}{s\beta-\lambda} \text{ and } R_m = \frac{S}{s\beta-(\lambda l + \lambda m)} \quad (19)$$

The total number of running instances committed to the elastic computing cloud service could be calculated using equation (19). The upper threshold utilization value is used for triggering to generate new instance in auto scaling mechanism. If the upper threshold value is 100%, then the provisioning new instance formula expressed as S.

$$S = \frac{R(\lambda l + \lambda m)}{R\beta-1} \quad S \leq Maximum\ Servers\ \&\ S \geq Minmum\ Servers \quad (20)$$

Another important performance metric is cost of the cloud. In cloud computing resources are adopt by hourly basis. In our queuing system follows the on-demand pricing model and this model considering computing resources, bandwidth of network usage and storage cost, thus the total cost of the queuing system derived as follows:

$$\text{Total Cost} = (P_{bw} \times \lambda_{GB/s} + P_{Com} \times S + P_{Sto} \times \lambda_{GB/s}) \times T \quad (21)$$

In Eq(21) P_{bw} is denoted as cost of the bandwidth of the network, P_{Com} is cost of the computing instance in the elastic cloud, P_{Sto} is cost of the storage server in terms of giga bytes , $\lambda_{GB/s}$ is the effective arrival rate in GB/s , S is the total number of servers and T is time in terms of hours.

3. Experimental model

To verified, analyzed and also compared the above analytical with the real world experimental results. Fig shows the experimental test-bed of hosting web application on AWS using auto scaling service. Main

components of this cloud datacenter are virtual machines are also called as servers, load balancer, RDS database, Route 53 and S3. An experimental architecture design has been discussed in [8].

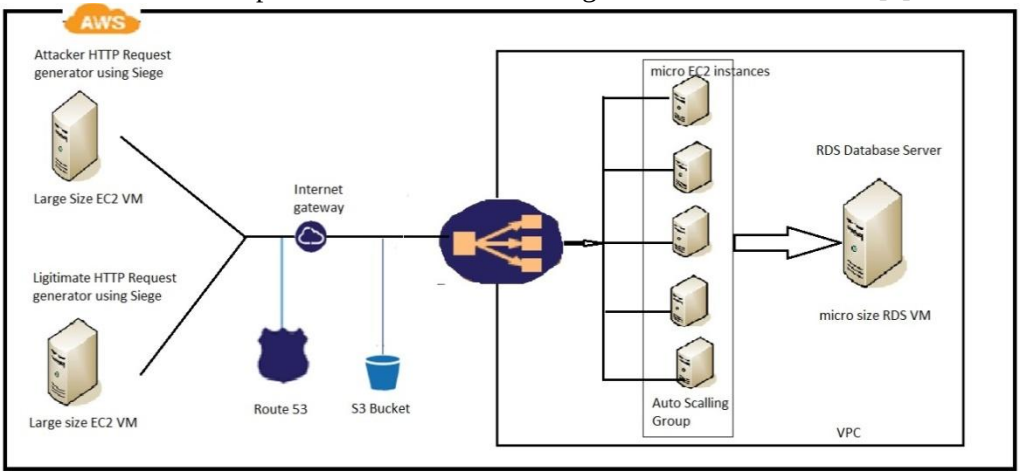

Fig 4: Web application hosting using elastic cloud on AWS

In the above figure micro size EC2 VM's are configured with auto scaling and for hosting web application [15]. RDS database server is used to store the network in and network out [16]; S3 bucket is created to store the client's logs [17]. Route 53 for to register domain name of the hosted web application in the EC2 [18] and final component of this architecture is large size EC2 VM's, it is used to generate http traffic of attacker and legitimate users.

For this analysis created micro word press EC2 instance in the EC2 console and attached my sql micro database server for storing the network traffic with multiple A-Z basis, this feature clones the database servers to all regions for availability purpose. For distributing the client's HTTP proxy to available VM's in the pool of VPC a load balancer can be configured and assigned auto scaling group with minimum two and maximum ten VM's with scale up and scale down conditions. We set up the Web server to restore a webpage page of a size of 580 bytes while tolerating a HTTP request from load balancer. The page measure was tuned until the point that we accomplished just about 100 % CPU use when the occasion gets 100 Req/sec (Requests each second) or, by the day's end uttermost scopes of our medium surveyed VM occasion. Regularly, this will give us around 10 ms for the run of the mill association time while modifying HTTP request a low rate. If it exceeds the 100 % CPU utilization of the VM then automatically generate a new instance and registered in the load balancer to distribute the requests at the same manner if it is below 30% of the CPU utilization then delete an instance from load balancer.

Attach configured load balancer to Route 53 for register domain name of the hosted web application. Finally launched large size instance and installed siege on that to generate legitimate and attacker http proxy. Ambush was planned to make HTTP request at the same time, and execution estimations were taken after a time of 15 min. For our estimation, we checked the running with execution estimations: response time, throughput, and CPU utilize. The run of the mill, scarcest, and most essential estimations for response time and throughput were given by the JMeter signify graph report toward the total of the run. With respect to CPU utilization, we used SAR Linux utility and Perl scripting tongue to gather and signify the CPU use readings of the running VMs. The CPU utilization readings were taken in the unfaltering time; particularly from 8 to 12 min.

4. Results

This paper for the numerical illustration assumed based on the size of the web application capacity of the micro virtual machine is 100 req/sec and it was examined by Catteddu and Hogben in 2009 [19]. This elastic cloud initially started with 2 servers. In the auto scaling provisioning overhead considered as 55.4 s, and it was taken from Islam et al. [20]. In our assumptions $\frac{1}{\alpha} = 50$ ms and $\frac{1}{\beta} = 20$ ms, legitimate users traffic 200 requests per second, attacker traffic varying from 200 to 2000 requests per second .

Total cost of the elastic cloud datacenter can be calculated using equation (22). As per the amazon aws price of the micro instances is $0.115, RDS server price is $0.115, base cost of $0.01 per GB in/out information exchanged dependent on the revealed costs of Internet information exchange "in" and "out" of Amazon EC2 [21].

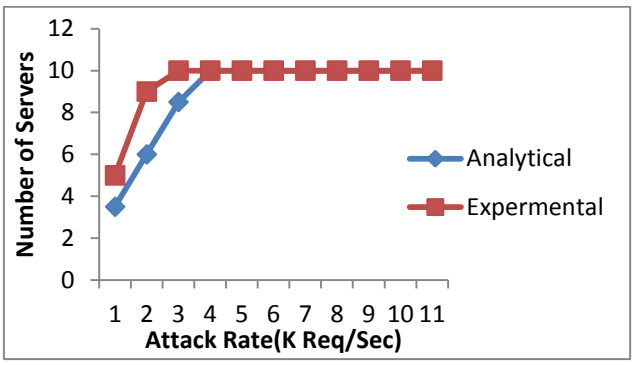

Fig 5: Total number of servers in relation to the attack rate

Figure 5 show the relation between attack rate and total number of servers running on the cloud. In the auto scaling configurations assumed least number of occurrences is two and greatest number of occasions is ten. Our architecture initially starts with two servers with legitimate traffic, if the attacker traffic occurs and incurred by the attacker then number of servers increased. When the client traffic exceeds the capacity of the cloud and touch the upper threshold value then number of server's value goes constant.

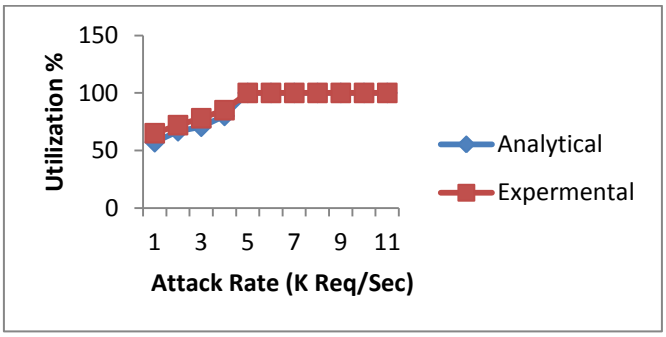

Fig 6: Elastic cloud utilization in relation to the attack rate

Figure 6 presents the utilization of the elastic cloud when it is incurred by the EDoS attack. The obtained results shows that regarding utilization, the attack rate of the hosted web application increased the corresponding utilization of the cloud also increased.

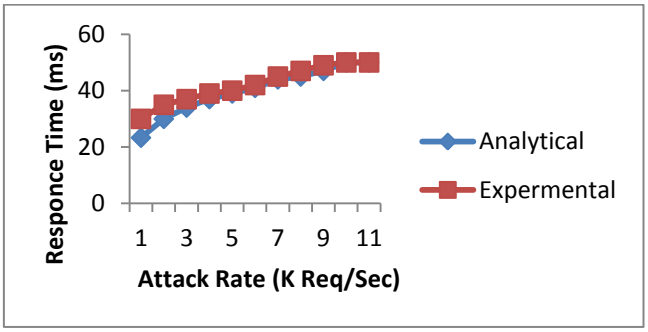

Fig 7: Traffic response time in relation to the attack rate

This paper assumed there is no waiting time of the queuing system because of auto scaling property. Figure 7 describes based on the obtaining results the attack rate expands, the relating reaction time additionally increments but there is no considerable variation when the attacker load goes high.

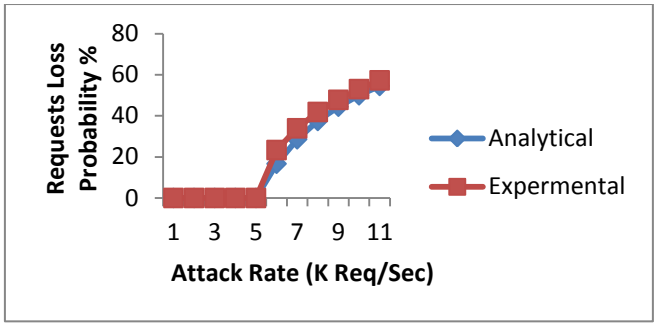

Fig 8: Requests loss probability in relation to the attack rate

Figure 8 depicts the curve of elastic cloud loss probability percentage. This figure shows that there is no loss probability at the light traffic occurs. The heavy traffic generated by the attacker, it utilize the cloud resources and generates the request loss after certain capacity of the cloud resource. At the point when the assault rate builds the comparing authentic users request loss also increases.

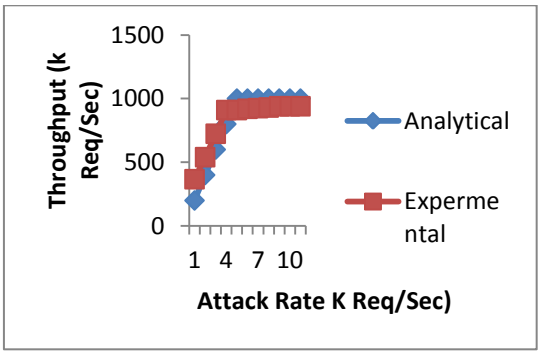

Fig 9: Throughput of the traffic in relation to the attack rate

Figure 9 presents the results of throughput of the cloud datacenter. Figure shows that the effective arrival rate of the incoming load. When the attack rate increases the throughput also increases, after exceeds the capacity of the cloud it goes to constant. Here the throughput of the legitimate user traffic also affected.

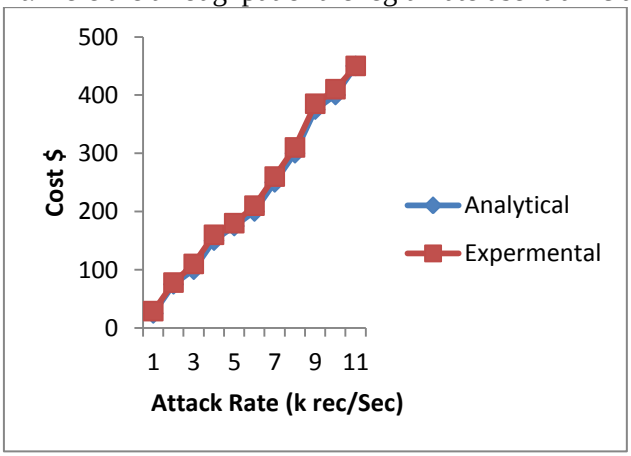

Fig 10: total cost of the elastic cloud in relation to the attack rate

Cost of the elastic cloud incurred by the EDoS attack shown in figure 10. As per the obtained results the attack rate increases corresponding cost of the cloud increases. This cost indicates the business loss of the vendors of the cloud.

5. Conclusions

This paper proposed a diagnostic model to think about the impact of the EDoS assault in the flexible distributed computing servers, thinking about number of execution measurements. These metrics are request

loss probability, total number of instance running on the elastic cloud, utilization of the cloud resources, request response time or latency of the request, throughput of the effective arrivals and total cost of the cloud setup. The loss probability and elasticity of the cloud and cost functions are evaluated. The obtained result shows that the performance metrics are incurred by EDoS attack. For loss probability, whereby legitimate request loss can goes high when high load spike. The results showed that unacceptable delay in end to end reaction time. The numbers of servers are increased and cloud utilization exceeds the capacity of the cloud. The results have shown there is little impact on the throughput. In addition cost of the cloud increased and it leads to economical loss of the business or cloud vendors. As a future work, propose an analytical model and experimental model to mitigate the EDoS attack and evaluate the cost of the cloud using different pricing models.

6. References

1. https://www.gartner.com/newsroom/id/3871416.
2. L. Wu and R. Buyya, "Service Level Agreement (SLA) in Utility Computing Systems", Technical Report, pp. 1-27, 2010.
3. Rashmi V (2015) et al. , Understanding DDoS Attack & Its Effect In Cloud Environment, Published by Elsevier, Procedia Computer Science 49 (2015) 202 – 210.
4. http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf
5. Hoff, C.: The economic denial of sustainability concept. http://rationalsecurity.typepad.com/blog/2008/11/index.html
6. Khaled Salah et al. (2015), "An Analytical Model for Estimating Cloud Resources of Elastic Services": Springer Journal.
7. F. Al-Haidari, et al. (2015). Evaluation of the Impact of EDoS Attacks Against Cloud Computing Services, (Arab J Sci Eng), Springer.
8. Suneetha Bulla, B. Basaveswara Rao, et al.: "An experimental evaluation of the impact of the EDoS attacks against cloud computing services using AWS", International Journal of Engineering & Technology, 7 (1.5) (2018) 202-208.
9. Khaled Salah," A Queueing Model to Achieve Proper Elasticity for Cloud Cluster Jobs", International Journal of Cloud Computing (ISSN 2326-7550),Vol. 1, No. 1, July-September 2013
10. Gian-Luca Dei Rossi, Mauro Iacono, Andrea Marin : "Evaluating the impact of eDoS attacks to cloud facilities": VALUETOOLS 2015, December 14-16, Berlin, Germany.
11. Shi, Y.; Jiang, X.; Ye K.: An energy-efficient scheme for cloud resource provisioning based on cloudSim. In: 2011 IEEE International Conference on Cluster Computing (CLUSTER), Austin, TX, pp. 595–599 (2011)
12. K. Salah, "Analysis of a Two-Stage Network Server," International Journal of Applied Mathematics and Computation, Vol. 217, No. 23, 2011, Elsevier Science, pp. 9634-9645.
13. https://books.google.co.in/books?id=Rw9DAQAAIAAJ&q=inauthor:%22Hideaki+Takagi%22&dq=inauthor:%22Hideaki+Takagi%22&hl=en&sa=X&ved=0ahUKEwjJh4OHz_XaAhXBk5QKHeyQBFwQ6AEIKjAB.21, Takagi, 1993.
14. Amazon web services, auto scaling. http://aws.amazon.com/autoscaling/
15. Amazon EC2: https://aws.amazon.com/ec2/.
16. Amazon RDS: https://aws.amazon.com/rds/.
17. Amazon S3: https://aws.amazon.com/s3/.
18. Amazon Rote 53: https://aws.amazon.com/route53/.
19. Catteddu, D.; Hogben, G.: Cloud Computing: benefits, risks and recommendations for information security. Technical Report, European Network and Information Security Agency (2009).
20. Islam, S.; Lee, K.; Fekete, A.; Liu, A.: How a consumer can measure elasticity for cloud platforms. Technical Report, School of Information Technology, Univercity of Sydeny (2011)
21. Amazon EC2 Pricing. http://aws.amazon.com/ec2/pricing/
22. P. Mell & T. Grance. "The NIST definition of cloud computing." 2011.