International Journal of Technical Research and Applications e-ISSN: 2320-8163,
www.ijtra.com Special Issue 31(September, 2015), PP. 261-267

A SURVEY OF DETECTION & MITIGATION MECHANISMS AGAINST EDOS ATTACK IN CLOUD ENVIRONMENT

Sukhada Bhingarkar
Ph.D. Student, Information Technology, Terna Engineering College, Navi Mumbai, India
sukhada.bhingarkar@gmail.com

***Abstract*— The Distributed Denial-of-service (DDoS) attack is considered one of the largest threats to the availability of cloud computing services which is used to deny access for legitimate users of an online service. But, Economic Denial of Sustainability (EDoS) attack is a special breed of DDoS attack that targets cloud's pay-as-you-go model. EDoS attack exploits auto scaling feature of cloud. The attacker generates malicious HTTP requests for web application. The Cloud Service Provider (CSP) scales the architecture automatically to service those requests for which cloud consumer is charged. This causes a sustainable decline in the economy of the consumer. The malicious HTTP traffic mimics to be legitimate and hence go undetected. As EDoS attack is carried over extended period of time, the security mechanisms against DDoS attack are not applicable to overcome EDoS attack. This paper presents an overview of detection and mitigation methodologies implemented so far against EDoS attack and it also points out research challenges in this field.**

I. INTRODUCTION

Cloud computing refers to delivery of computing resources over the internet. Cloud computing provides shared pool of resources (example: networks, memory, computer processing, user applications) that can be rapidly provisioned and can be put out with minimal exertion. There are several benefits of cloud computing, such as cost savings, scalability, reliability, maintenance, mobile accessible etc. Besides all these benefits, Cloud Computing does come at the cost of increased security risks which is currently one of the biggest challenges this technology is facing today, limiting the number of organizations willing to embrace it wholeheartedly. DDoS is one type of aggressive attack which causes serious impact on cloud servers. According to [1], in the past year, there has been a 22% increase in total DDoS attacks, and a whopping 72% increase in average attack bandwidth.

A. Distributed Denial of Service (DDoS) Attack

A Distributed Denial of Service (DDoS) attack is a malicious attempt to make a server or a network resource unavailable to legitimate users by overloading the server with large number of requests. In DDoS attack, attacker begins by gaining the control of initially one computer and treats it as DDoS master. Then, it gains illegal access to as many computers on Internet as possible and DDoS master instructs these compromised machines to send a flood of requests to the target server. The target server eventually gets overwhelmed and starts denying the requests of legitimate users [2].

1) *DDoS attack on Web 1.0 Applications:* Web 1.0 is the first generation of the web which can be considered as read only web which involves limited user interactions or content contributions. It consists of static web pages and only allows searching the information and reading it [3]. Example of web 1.0 is shopping cart application. If such application is hosted over the cloud, then two types of DDoS attack are carried out over such website. The first takes place at the network layer (Layer 3 and 4) and the second at the application layer (Layer 7). At the network layer, attack brings down a website by overwhelming network and server resources, causing downtime and blocking responses to legitimate traffic e.g. UDP Flood, ICMP Flood and Ping of Death. Application layer DDoS attacks mimic legitimate user traffic and crash the web server by searching for content on the site or clicking the "add to cart" button. e.g HTTP flood.
2) *DDoS attack on Web 2.0 Applications:* Web 2.0 is the second generation of World Wide Web that is focused on the ability for people to collaborate and share information online. Web 2.0 basically refers to the transition from static HTML Web pages to a more dynamic Web that is more organized and is based on serving web applications to users. Examples of Web 2.0 include social networking sites, blogs, wikis, video sharing sites, hosted services, Web applications, and web mashups [4]. A mashup is a web application that uses content from more than one source to create a single new service displayed in a single graphical interface. The architecture of web mashup is shown in figure 1 [5].

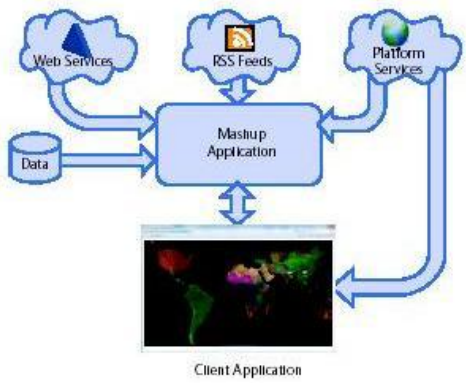

Fig. 1. Web Mashup Architecture

When web mashup application is hosted over cloud, it is threatened by many attacks, one of which is DDoS attack. The mashup application may become a target of DDoS attack as follows:

i. Multiple client applications may be a botnet that mimics a legitimate web browser and tries to overwhelm the bandwidth of mashup application by a flood of HTTP requests.

ii. A DDoS attack is possible whenever third party Javascript is executed within client's browser. Third party Javascript logic can include a loop that repeatedly requests resources from targeted mashup application.

Cloud Computing follows utility model where users are charged based on the usage of the cloud's resources. This pricing model has transformed the Distributed Denial of Service (DDoS) attack problem in the cloud to a financial one known as Economic Denial of Sustainability (EDoS) attack [6].

This paper describes the EDoS attack and the methodologies implemented so far for the detection and mitigation of EDoS attack.

The rest of the paper is structured as follows: section II briefs about EDoS attack and the difference between DDoS and EDoS attack. Section III discusses the detection methodologies applied to differentiate botnet and legitimate users. Section IV presents mitigation techniques implemented to lessen the effect of an attack. Section V gives brief summary and analysis of all the techniques. Section VI focuses on research challenges and concludes the paper.

II. Economic Denial of Sustainability (EDoS) Attack

DDoS attacks in traditional networked (non-Cloud) environment usually disrupt the service which hurts reputation and incurs economic loss. In Cloud environments, disrupting a service is not so easy due to its inherent capability of auto-scalability and service level agreements (SLA).

However, DDoS attempts on Cloud environments have another more alarming repercussion in that it does the consumption of more Cloud resources to provide auto-scalability, which normally exceeds the economic bounds for service delivery, thereby incurring Economic Denial of Sustainability (EDoS) for the organization whose service or Virtual Machine (VM) is targeted.

EDoS is a new breed of DDoS attack specific to Cloud environments. In this kind of attack, the Cloud service provider activates more and more resources to meet the SLA for the availability of the service for the customer, which eventually adds extra billing cost leading to EDoS. Cloud resources are metered on resource billing. Hence, the fraudulent consumption of bandwidth and computational resources of Web based cloud services incurs financial burden on the Cloud consumer and thus exploits cloud utility model.

A. EDoS Attack Threat Model

The target of EDoS attack is a public-facing web application or website hosted in a public Cloud Service Provider environment that is governed by a utility compute pricing model. In this kind of attack, the attacker's intention is not to make the cloud service unavailable but to put financial burden over cloud consumer by consuming metered bandwidth of web application hosted over cloud.

The following actors are involved in EDoS attack:

i. Cloud Service Provider (CSP): rents its resources and performs billing

ii. Cloud Consumer: uses cloud resources to host its web application

iii. Legitimate Client: accesses the services provided by cloud consumer.

iv. Attacker: intentionally generates fraudulent traffic to hit the economy of cloud consumer.

The EDoS attack network model is shown in figure 2.

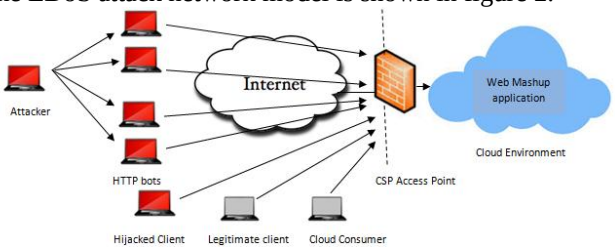

Fig. 2. EDoS Attack Network Model

Now days, the web sites are based on web 2.0 architecture. Hence, assuming that web mashup application is hosted over the cloud, EDoS attack can be performed as follows:

1. The attacker generates HTTP requests by forming a distributed botnet on the internet. These requests have a heavy workload effect on the hosting web application e.g. requesting large files, making frequent searches on entire product range, which results in large queries on backend databases such as involving the joining of tables. The workload can be on any of the resources; bandwidth, processing, memory etc. In this case, the CSP activates more and more resources to meet the SLA for the availability of the service for the customer, which eventually adds extra billing cost leading to EDoS.

2. In case of E-commerce applications, the cloud consumer earns profit if the client makes a purchase. But if the client

only browses the site without making a purchase, then the consumer earns no profit but in turn pays to the CSP.

3. EDoS attack is also possible through hijacked client browsers specifically in web 2.0 architecture wherein malicious code can be injected in the browser by the hacker that generates repeated requests to targeted mashup application resulting in overwhelming the bandwidth of mashup application.

B. Difference between DDoS and EDoS Attack

There are two major differences between EDoS and DDoS attacks. First, EDoS attacks aim to make cloud resources economically unsustainable for the victim, whereas DDoS attack aims to degrade or block cloud services.

Second, DDoS attacks are carried out in a short time period whereas EDoS attacks are more subtle and carried out over a long period of time.

Third, EDoS attack occurs just above the normal activity threshold and below the DDoS attack threshold [7].

Therefore, it may be unlikely to be detected by traditional intrusion detection systems and also the methodologies used to overcome application layer DDoS attacks are not applicable to EDoS attack.

III. EDOS DETECTION METHODOLOGY

Detecting EDoS attack is very difficult because the way an attacker requests web resources is like that of any legitimate client and the only differentiating attribute is their intention [7]. Thus, the purpose of detection methodology is to differentiate bot behaviour from human behaviour.

A. Zipf's Law Distribution

Zipf's law was originally introduced in the context of natural languages and is performed by calculating the frequency of occurrence F of each word in a given text. By sorting out the words according to their frequency, a rank R can be assigned to each word, with for the most frequent one [8].

The methodology discussed in [9] applies the properties of Zipf's law in the analysis of aggregated user consumption patterns. The web server log contains request record for the web pages. Let fi be the frequency of requests and i be the rank assigned to the page. The page which is referred most is assigned rank one and so on. Thus, if Zipf's law holds, then the frequency fi is inversely proportional to the rank of the page. A typical Zipf's Law rank distribution is shown in Figure 3. The y-axis represents occurrence frequency, and the x-axis represents rank (highest at the left) [8].

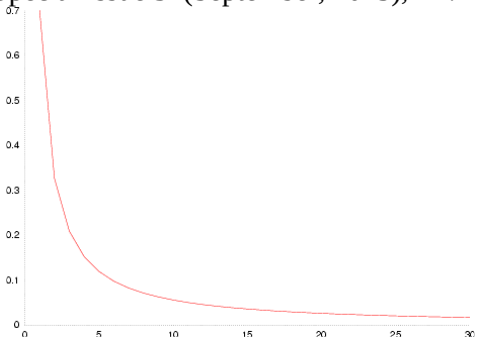

Fig. 3. A typical Zipf's Law rank distribution

The detection methodology discussed in [9] determines Zipf's distribution for the training and test data sets and then computes linear regression line for each distribution. The slopes of the respective linear regressions are compared with the statistical hypothesis that the slopes are the same. If analysis indicated that the slopes were significantly different, then, it concludes that fraud motivated access patterns are observed in Web request logs.

B. Entropy Detection

In order to detect EDoS/ FRC attack, individual user behaviour is modelled in [9] by analysing the entropy of session lengths generated by an individual over a fixed duration of time. The session length is defined as the number of web documents requested during a session. The entropy of session length for session j is Hj composed of the n events is defined as:

$$H_j = - \sum p_i log_2(p_i)$$

The hypothesis in [9] is that, randomly generated session lengths, deviate sufficiently from a profile of normal user behaviour. The proposed detection methodology in [9] computes a standard of entropy of normal session lengths based on a Web request log and then calculates entropy of session lengths for each unique user. Then, it compares the entropy result to the standard. If a user's session length entropy is outside the standard, the user is designated as malicious.

C. Time Spent on a Page (TSP) based Detection

The methodology in [10] considers Time Spent on a Web Page (TSP) as a request dynamic to detect the attack traffic. As the bot traffic is automatically generated and its intention is to create heavy workload by browsing the web pages, a large traffic having small TSP values can be considered as malicious.

Firstly, TSP is calculated for each page as the difference between timestamps of two consecutive requests for web pages. Then, the mean of TSP is calculated from the data set Xi which represents the page requests for a page i. The requests generated by each cloud user are collected in form of logs at cloud controller. Each user has a separate log. The deviation of each log request from mean TSP is calculated.

Lastly, Mean Absolute Deviation (MAD) is determined by taking the average of these deviations.

For each user, MAD plot is drawn showing deviation vs page visited and compared it with the mean of all pages. If the curve of any user deviates more from the mean curve, then that user is identified as malicious.

D. Web Usage Features based Detection

The application layer DDoS attacks focus on request rates of clients to differentiate between legitimate user and attacker. But, this technique is not applicable to detect attackers in EDoS attack as EDoS attack sustains over a long period of time. Hence, the attribution methodology presented in [11] targets four aspects of client web browsing behaviour i.e. request volume, session volume, average session length and chi-square statistic.

The quantity of primary requests invoked by a client within an observation time period is called as Request Volume. Primary request is explicit request from a client to whereas secondary request originates from primary web document to retrieve certain image, video etc. The minimum threshold considered is 5 requests per client according to Cumulative Distribution Function (CDF) calculated over training data set. As the intention of attacker in FRC attack is to consume as much bandwidth as possible, the attacker generates number of requests more than the threshold.

Session Volume is the quantity of web sessions attributed to a single client within an observation period. The attacker in FRC attack distributes his/her resource consumption over the course of many days by launching multiple fraudulent web sessions. According to CDF for training data set, the average session volume per client is three. Thus, the client requesting slightly more than three sessions is flagged as malicious.

The number of primary requests in a web session is termed as session length and the mean of these lengths is called as average session length. According to CDF, the average session length is considered to be less than that of 5 requests. The attacker usually tries to keep average session length minimal, but for that the attacker is forced to initiate more web sessions which increases his/her session volume score and hence can be identified as fraudulent.

The Zipf like distribution for training data set broadly states that 10% of the requested documents are requested 90% of the time i.e. a significant fraction of normal client behaviour is reasonably self-similar to the overall client population. Thus, chi-square statistic is used as a relative measure of similarity or dissimilarity between individual client request distributions and the overall population distribution.

The web pages are ranked and are grouped into discrete bins, each bin having probability π. The expectation for each bin is $E_i = n\pi_i$. For each client in the test dataset, a chi-square statistic is computed as:

$$\chi 2 = \sum (n_i - E_i)^2 / E_i$$

Then, the overall CDF is constructed. The client having high chi-square statistic score is considered as legitimate.

The scores for all above four metrics are summed together and compared against threshold to identify client as legitimate or malicious.

IV. EDOS MITIGATION METHODOLOGY

Mitigation techniques are applied to lessen the effect of an attack. This section describes various frameworks proposed so far to mitigate EDoS attack.

A. EDoS Armor

The mitigation technique called as EDoS Armor in [12] concentrates on protecting E-commerce applications with the assumption that attacker performs EDoS attack by not following regular workflow of E-commerce applications by purchasing the item but, by idle surfing of the web sites for entertainment or price checks.

EDoS Armor proposes muti-layered defence system which includes two modules:

(i) Admission control: Challenge Server identifies whether the request is from legitimate client or bot by means of challenge which can be either of image based or any cryptographic challenge. Thus, it authenticates number of users in the system. Then, it allows limited no. of valid clients to send the requests simultaneously through port hiding. This avoids over burdening.

(ii) Congestion control: This module takes care that maximum resources are available to the good clients. The client is categorized as good or bad client depending upon his browsing behaviour. Decision tree algorithm J48 is used over access log to classify the clients. The parameters used for classification are like Purchasing History, CPU Processing time, Session information, Resources Access Pattern. The priority is assigned to the clients based on types of resources they visit and the types of activities they perform.

B. In-cloud Scrubber Service

In-cloud scrubber service is an on-demand service proposed in [13] to mitigate network layer and application layer EDoS attack based on puzzle approach. The cloud service is switched between two modes: normal and suspected based on server load and network bandwidth. If the resource depletion level goes beyond the limit and bandwidth traffic is also very high, then the service provider suspects high rate attack. The system switches to suspected mode and called scrubber service.

The primary function of scrubber service is to generate a puzzle to check legitimacy of the client. The service generates partial hash input and hash output and transmits both pieces of information to the client.

$$\text{i.e. } H(X \,||\, k) = Y$$

where, X and Y are the puzzle parameters provided by service and k is a puzzle solution.

The client is supposed to apply brute force method to find out the value of k. Here, as the puzzle generation and verification is done by third party i.e. scrubber service, the burden on Cloud Service Provider is reduced.

C. sPoW (Self Verifying Proof of Work)

The idea behind mitigation technique presented in [14] is to grant access to only those clients who are willing to pay for the service. The client has to contact sPoW Name server for name resolution when it wants to establish communication with the server. The client defines crypto puzzle difficulty level, k and subsequently makes a request. The server in turn generates a puzzle of required difficulty level which client is supposed to solve.

If an initial connection request is not successfully made during a given frame of time, the client may request for a more difficult puzzle. Upon successfully solving the puzzle of given difficulty level, the server establishes a secure communication channel for message exchange. This method transforms network level EDoS into traffic which can be distinguished through basic packet pattern matching. It helps to discard attack traffic before billing is triggered.

Also, application level EDoS is addressed by prioritizing the traffic. The existing connection requests have given priority over initial connection requests. The existing connection which carries purchase transaction traffic is assigned more priority than casual browsing traffic carrying connection. The initial connection requests are prioritized using sPoW scheme in which priority is based on the resources expended by client in solving the puzzle because this reflects the urge of client to establish the connection. The sPoW is self-verifying because crypto-puzzle consists of both server channel connectivity details and partial encryption key. By brute forcing k bits, client discovers server channel and can place initial connection request which is queued at server end based on difficulty level of puzzle. Thus, this technique removes the requirement of separate verifier and ensures legitimate client request as client had expended enough resources to establish the connection.

D. DDoS Mitigation System (DDoS-MS)

The mitigation mechanism proposed in [15] is applicable to those enterprises which allow their employees to bring their own mobile device at the workplace to access enterprise database. This policy termed as Bring Your Own Device (BYOD) results into the threat of EDoS attack as the devices used by the employees are not configured by the organization.

The idea behind DDoS-MS framework is to test first two packets of each session in two successive stages instead of testing all the packets. First packet is tested by verifier node using Graphical Turing Test. Second packet is tested by client puzzle server which uses crypto puzzle to verify the source of packets.

The firewall adds the source IP address of incoming packet in either white list or black list depending on the result of verification process. The green nodes hide location of the server. The server receives only those packets which come through the green nodes. DDoS-MS enhances EDoS shield framework by decreasing end-to-end latency.

E. Cloud Trace back Model (CTB)

Mary et.al. [16] have proposed a combined approach to protect the cloud against DDoS and EDoS attack. Cloud Trace Back architecture applies SOA to trace back methodology to identify true source of DDoS attack. CTB is based on Deterministic Packet Marking Algorithm wherein the attacker sends SOAP request message for web service to CTB. CTB places Cloud Trace Back Mark (CTM) within the header. Then, the SOAP message is sent to the web server. Upon discovering of an attack, the mark can be extracted to reconstruct the path.

CTB does not eliminate DDoS attack. Hence, trained back propagation neural network model is used called as Cloud Protector.

EDoS attack is detected using Verifier nodes which are a pool of virtual machine nodes. V-nodes verify the legitimate requests at application level using unique Turing test e.g. unique Question Testing. Depending upon the result of verification, the source IP address of the request is added to the white list or black list which is maintained by Virtual Firewall. Here, the architecture assumes that the system is protected against IP spoofing attacks.

V. Review of Countermeasures

A survey carried out on detection and mitigation methodologies against EDoS attack show that there are still opportunities present to carry out further research work. Table I and II shows the summary of detection and mitigation techniques along with their limitations which can lead to further research.

TABLE I
Analysis of Detection Techniques

Sr. No.	Approach	Methodology	Advantages	Limitations
1	Exploiting Cloud Utility Models for Profit and Ruin	- Zipf's law Distribution - Entropy Detection of Session Lengths	Effectively detects anomalous behaviour from web request logs.	- Supports only SaaS kind of service. - Does not consider Web 2.0 architecture
2	Detection of Economic Denial of Sustainability using Time Spent on a	Calculation of Mean Absolute Deviation of Time Spent on Web Page	Simple method to differentiate legitimate traffic from attack.	- Supports only SaaS kind of service. - Does not consider Web

	Web Page in Cloud			2.0 architecture
3	Attribution of Fraudulent Resource Consumption in the Cloud	Analysis of web browsing behaviour. - Request volume - Session Volume - Avg. Session Length - Chi-square statistic.	Minimum False Positive and False Negative Rate.	- Attacker can learn normal request patterns. - Does not consider Web 2.0 architecture

TABLE II
ANALYSIS OF MITIGATION TECHNIQUES

Sr. No.	**Approach**	**Methodology**	**Advantages**	**Limitations**
1	EDoS Armor: A Cost Effective Economic Denial of Sustainability Attack Mitigation Framework for E-Commerce Applications in Cloud Environments	- Authentication through Crypto-puzzle - Port hiding - Decision Tree algorithm	Classifies users effectively. Supports dynamic web applications	- Provides defense only for E-commerce applications. - Does not consider Web 2.0 architecture
2	Mitigating Economic Denial of Sustainability in Cloud Computing using In-Cloud Scrubber Service	- Authentication through Crypto-puzzle	On-demand Scrubber service which removes the burden from CSP.	- Legitimate user is unwilling to solve such puzzles. - Prevents only network level EDoS attacks.
3	sPoW: On-Demand Cloud based eDDoS Mitigation Mechanism	- Crypto puzzle - Packet Filtering	Prevents EDoS traffic from using costly cloud resources due to the provision of self-verification.	- Prevents only network level EDoS attacks.
4	A New Method to Mitigate the Impacts of Economic Denial of Sustainability Attacks against the Cloud	- Graphical Turing Test - Crypto puzzle	Tests only first two packets rather than testing all the packets. Decreases end-to-end latency.	- Does not deal with IP packet fragmentation. - Does not deal with dynamic IP addresses.
5	Secure Cloud Computing Environment against DDoS and EDoS attacks	- SOA applied to Cloud Trace Back Model - Neural Network - Unique Turing Test - Packet Filtering	Combined approach against DDoS and EDoS attack	- Does not deal with IP spoofing.

VI. RESEARCH CHALLENGES

There are following issues in present detection and mitigation approaches:

- All present approaches concentrate only on differentiating user as legitimate and or malicious (i.e.botnet). However, today's websites are based on web 2.0 architecture. Hence, if website hosted over a cloud is using mashup technology, then it may receive requests not only from users but also from another website. For example, if the website is of travelling website, it in turns calls API of another web site like hotel booking. So current approaches lack to distinguish whether the request is generated from bot or it is from another website.
- Web mashup application can be targeted by hijacked client's browsers that execute malicious code to generate repetitive HTTP requests. This threat is not considered in the current approaches.
- Most of the current approaches use CAPTCHA test to differentiate between human and bot. But, legitimate users are unwilling to solve CAPTCHA test. Also, bots are able to solve CAPTCHA test.
- As a part of mitigation strategy, most of the approaches deny the request once the user is classified as malicious but this is less elegant solution.

VII. CONCLUSION

EDoS attack is more subtle attack than DDoS that seeks to disrupt long-term financial viability of operating in cloud by exploiting utility pricing model.

Unlike short lived DDoS attacks, EDoS attacks span over a long period of time. Hence, traditional DDoS defense techniques are not applicable to defend EDoS attack. Until recently, this attack is not much addressed. The current approaches discuss threat model and defense strategies for web contents hosted on cloud which follow web 1.0 architecture. But in an age of browser-based botnets and web 2.0 applications, it is necessary to build up multi layered framework which can do traffic profiling and visitor classification effectively.

REFERENCES

[1] DDoS Attacks Evolve into Sophistication, http://www.infosecurity-magazine.com/news/ddos-attacks-evolve-into/ [Online; accessed 22-Jul-2014].

[2] Shui Yu et.al., "Can We Beat DDoS Attacks in Clouds?", IEEE Transactions on Parallel and Distributed Systems, Vol. 25, No. 9, September 2014, pp. 2245-2254

[3] [Online]. Available: Web 1.0 http://computer.howstuffworks.com/web-101.htm.

[4] James Governor, Dion Hinchcliffe, Duane Nickull, "Web 2.0 Architectures", O'Reilly, 2009.

[5] [Online] Available: Enterprise Mashups, https://msdn.microsoft.com/en-us/library/bb906060.aspx

[6] J. Idziorek, M. Tannian, and D. Jacobson, ''Insecurity of Cloud Utility Models,'' IT Prof., vol. 15, no. 2, pp. 22-27, Mar./Apr. 2012.

[7] Saeed Shafieian et.al. "Attacks in Public Clouds: Can They Hinder the Rise of the Cloud", Cloud Computing: Challenges, Limitations and R& D Solutions, Springer, 2014.

[8] Shi-Ming Huang et.al. "An Investigation of Zipf's Law for Fraud Detection", Decision Support Systems, Vol. 46, Issue 1, Dec. 2008, pp-70-83.

[9] J. Idziorek and M. Tannian, "Exploiting Cloud Utility Models for Profit and Ruin," Proc. 2011 IEEE 4th Int'l Conf. Cloud Computing (Cloud 11), pp. 33–40.

[10] Anusha K. et. al. "Detection of Economic Denial of Sustainability using Time Spent on a Web Page in Cloud", IEEE International Conference on Cloud Computing in Emerging Markets (CCEM), 2013

[11] J. Idziorek, M. Tannian, and D. Jacobson, "Attribution of Fraudulent Resource Consumption in the Cloud," Proc. 2012 IEEE 5th Int'l Conf. Cloud Computing (Cloud 12), IEEE, 2012, pp. 99–106.

[12] Muddassar Masood et.al "EDoS Armor: A Cost Effective Economic Denial of Sustainability Attack Mitigation Framework for E-Commerce Applications in Cloud Environments", IEEE International Multi Topic Conference (INMIC)Conference, 2013

[13] M. Naresh Kumar, et. al.. "Mitigating economic denial of sustainability (edos) in cloud computing using in-cloud scrubber service", In Proc. of the Fourth Intl' Conf. on Computational Intelligence and Communication Networks (CICN), 2012.

[14] Soon Hin Khor and Aki Nakao. "spow: On-demand cloud-based eddos mitigation mechanism", In In Proc. of the Fifth Workshop on Hot Topics in System Dependability, 2009.

[15] Wael Alosaimi and Khalid Al-Begain, "A New Method to Mitigate the Impacts of Economic Denial of Sustainability Attacks Against the Cloud", 2013

[16] I. Mettildha Mary et. al. "Secure Cloud Computing Environment against DDoS and EDoS attacks", International Journal of Computer Science and Information Technologies, Vol. 5, 2014