Economic Denial of Sustainability Attacks Mitigation in the Cloud

Wael Alosaimi[1], Michal Zak[1], Khalid Al-Begain[1], Roobaea Alroobaea[2] and Mehedi Masud[2]

[1]University of South Wales, United Kingdom
[2]College of Computers and Information Technology, Taif University, Saudi Arabia,

***Abstract*:** Cyber security is one of the most attention seeking issues with the increasing advancement of technology specifically when the network availability is threaten by attacks such as Denial of Service attacks (DoS), Distributed DoS attacks (DDoS), and Economic Denial of Sustainability (EDoS). The loss of the availability and accessibility of cloud services have greater impacts than those in the traditional enterprises networks. This paper introduces a new technique to mitigate the impacts of attacks which is called Enhanced DDoS-Mitigation System (Enhanced DDoS-MS) that helps in overcoming the determined security gap. The proposed technique is evaluated experimentally and the result shows that the proposed method adds lower delays as a result of the enhanced security. The paper also suggests some future directions to improve the proposed framework.

***Keywords*:** Cloud Computing, DDoS, Denial of Service, Distributed Denial of Service attacks, DoS, Economic Denial of Sustainability, EDoS.

1. Introduction

The word security is widely used with cloud computing. The outcomes of the survey that is carried out by [1] shows that almost 90% of respondents are concerned with the security of the cloud. The cloud computing industry is mostly influenced by the users trust on the available security measures that can safeguard their services and data.

The security issues related to the cloud computing include virtualization issues, privacy breach, and specific legal challenges. The significance of these issues cannot be neglected for the acquisition of a confidence of the participants as they will not be worried for their protection while existing in the cloud environment.

Availability is a crucial element and mostly targeted by the attackers in cloud computing. Availability is considered equally with the security in cloud computing as its clients need to get the same accessibility to their data on the cloud as if it is in their local machines.

Despite of other security issues that will be mentioned in this study, the availability challenges linked with cloud computing will get more attention. The availability is more open to threats such as the Denial of Service (DoS), Distributed Denial of Service (DDoS), and the Economic Denial of Sustainability (EDoS) attacks and their detailed explanation will be presented in the next sections of this study. Moreover, there is information provided about the principles of the attacks, launch of such attacks, and their variants. The existing mitigation solutions that are proposed to protect the availability are evaluated in terms of their strong aspects and their limitations.

This paper enhances the previous work of the authors [2] by evaluating the Enhanced DDoS-Mitigation System (Enhanced DDoS-MS) performance in order to prove its effectiveness in protecting the targeted system with low response time for the legitimate users.

The problems that can occur in the future times are presented through three standpoints in the last section of this paper. These are:

- Arising security issues in the context of cloud computing.
- Arising issues in the context of DDoS (Distributed Denial of Service attacks).
- Arising issues in the Enhanced DDoS-MS framework context

Cloud computing refers to the computing model in which delivery of applications and services to the end-user are done through the internet as an on-demand service. All these services and applications are delivered to the clients by means of the Cloud Service Providers (CSPs) that own and control huge data centers all over the world. These data centers have high-grade servers that interlinked together to form the cloud that hosts web servers and web applications [3]. The extensively attractive characteristics of cloud include elasticity, flexibility, scalability, and its availability.

These specified characteristics enable the clients of the cloud to acquire advantages unswervingly when they subscribe to the cloud. Such advantages include increasing storage capacity, cost reduction, reduction of the IT relative issues.

The cloud services are presented to the clients based on their types in three categories. These are public cloud, private cloud and hybrid cloud. Furthermore, the subscription of services of cloud can be at various levels which are Infrastructure as a Service (IaaS), Software as a Service (SaaS), and Platform as a Service (PaaS). Nevertheless, cloud clients and providers face a potential threat of security which is elaborated in this section. Such security concerns are classified into four categories that are described below [4]:

- Policy and Organizational Risks including compliance risk, loss of control, end of service, and portability issue.
- Physical Security Issues.
- Technical Risks such as encryption issues, Network Attacks including Man in the Middle Attack (MITM), Distributed Denial of Service (DDoS), port scanning, IP spoofing , service outages, virtualization vulnerabilities, job starvation issues, data level security, web application security issues, data segregation, multi-tenancy security.

- Legal Issues such as data breach, data location, Data Deletion, and contracts designing and commitment.

a) Denial of Service (DOS)

In this section, we are going to talk about the variants of DoS attacks, their amplified versions, and their mitigation methods. The DoS can be generated to affect networks through different layers such as network, transport, or application layers. The significance of availability cannot be ignored as it is an important characteristic of any network or service. The flooding or Denial of service (DoS) attack harms this significant feature by prohibiting the legitimate customers from accessing the network resources. To serve the purpose of consuming servers processing power and the network capacity (bandwidth), the attackers commence producing DoS attacks by means of transmitting countless requests so the legitimate users become incapable of accessing the network even they are eligible for legitimate access to the network resources [5].

The protection of cloud is significant. It needs to be confined from three types of intimidations that floods the web page. These types are utilizing the system resources that affects the computational capacity, downloading large files from the web server which influences its communication capabilities and the bandwidth, and using password guessing attacks and SQL injections [6]. The attack stream against a static web page are launched through bonnets, computer viruses, or other available denial of Service tool. The flood might be harmful as a Denial of Service attack or normal event like flash crowd phenomenon. [7] defined the flash crowd as an event of utilizing a famous and known website by a very huge number of users simultaneously which causes a rush in traffic that renders the website inaccessible.

The differentiation between the Denial of Service and the flash crowd is significant. The Denial of Service results from large amount of requests that are suddenly demanded by a tiny set of known and new clients whereas the flash crowd results from large amount of requests that are demanded by a large number of legitimate clients after a specific social occasion [7]. This section focuses only on the Denial of Service (Dos) phenomenon.

b) Distributed Denial of Service (DDOS)

Distributed Denial of Service (DDoS) attacks are DoS attacks that are launched by distributed sources simultaneously. To protect the network from such threats, many security measures have been developed including:-

1. Filtering methods (based on attack patterns or threshold value)
2. Overlay-based mitigation techniques (employing distributed firewalls and concealing the defended server's location)
3. Trace-back methods (marking the malicious packets and trace their origins)
4. Push back methods (applying the process of filtration by the routers near the sources)

Nevertheless, the current solution limitation lies on the absence of the source's verification or the increase of the response time for the benign users. The computers that are employed in the flooding attacks are mostly infected by worms so their owners are unaware that they are a part of a malicious attack. The intention of the attacker is to generate a network of computers that is fully controlled by him to ensure the achievement of his attack. The attacker commences with penetrating the victims computers in order to create backdoors on them so he would easily manages them for a long duration. These penetrated computers are commonly known as bots or zombies. The attacker successfully manages the bots he can generates several kinds of attacks. One of these attacks is the DDoS. [7, 8, 9]

c) Economic DDOS

In the modern era of cloud computing, a new sort of DDoS attacks known as Economic Denial of Sustainability (EDoS) came into being and was presented by [10]. EDoS can be defined as the "packet downpour that stretches the suppleness of metered-services engaged by a server For example; a cloud based server".

An EDoS attack can be launched by distantly motivating bots to overwhelm the targeted cloud service utilizing bogus requests which are concealed from security breach detectors. Hence, the cloud service will be provided to the requester in an on-demand basis in a scalable manner. The cloud is highly dependent on pay per use base so that the customer's bill will be charged for these bogus requests, forcing the client to depart from the cloud service [11].

The most of the dreadful impact of such drawbacks will be resulting in loss of clients of cloud computing as they will prefer choosing an inexpensive and more effective mean to handle their business on their premises and data center instead of the cloud that charges them for unreal requests [12][13].

EDoS attack is actually an enhanced economic version of the Distributed Denial of Service (DDoS) attack with distinctive and more dangerous effect as the DDoS attacks majorly flood the targeted servers and create a huge amount of traffic in it which makes the network inaccessible for legitimate users. It is a great challenge for DDoS attack to be succeeded in harming the cloud as it has a large pool of resources, but the adversaries can launch their attacks against the weaker side which is the cloud customer's network. In such scenario, a vast quantity of bogus requests will be sent to the client's system which is successfully served by the cloud provider according to a specific contract that ensures responding to the high demands of the customer by scaling up the requested infrastructure. This process will charge the customer's bill. Therefore, in customer's opinion the cloud is highly expensive and unaffordable. The provider's profit can be affected by spreading the same feeling among many clients.

The authors of [14] divided the network security attacks into two sorts which are destructive and highly expensive. Comparatively, it is obvious that the DDoS attack is a destructive attack whereas the EDoS is extremely expensive.

As a result of the above discussion, the solution for the EDoS attacks must be a proactive solution. This means that it must be implemented in the customer's network in order to protect it from DDoS attacks and protect the provider from EDoS attacks.

Many methods are proposed to solve this issue. However,

these techniques are either verifying all received packets from a source which causes latency or sometimes verifying the first packet only excluding any more tests, which is not adequate to defend the system. The concept of limiting the response time is very significant as well as offering a strong protection against the destructive attacks. The authors of [14] actually persuaded on the significance of such concept as the organizations must offer a balance between providing security and their customers' convenience. It is under the cloud concept to designate a threshold value for the customer's usage in order to protect their bills as the cloud services must be scalable, elastic, metered by utilization, has shared pool of resources, accessible through the internet, and provided as an on-demand self-service according to the most agreed definition of the cloud service which is the NIST definition [15].

2. Related Work

The countermeasure approaches of Distributed Denial of Service (DDoS) are branched into two types i.e. proactive and reactive [15] :

I. Proactive Approach: it involves the solutions like overlay-based approach that treats the data packets before they access the protected system as they have the filters along with the other mechanisms. Furthermore, they are highly reliant on the nodes or distributed firewall for the purpose of concealing the protected server's location [16, 15].

II. Reactive Approach: such as the filtering mechanism that endeavors to alleviate the attack after its arrival to the protected system. These approaches are in question due to their precision in differentiating the legitimate packets from the malicious ones along with their effectiveness in formulating a profound filtering system lessening the impact of attacks in the targeted server [17].

There are certain drawbacks of the filtering approaches as stated below [15]:

a. Once an attacker penetrates the account of a legitimate user, the user's IP address can be used to access the system files and to cause harm to it.

b. The filtering systems work by first identifying the statistical anomalies or the known attack patterns. The problem is that these patterns and anomalies can be easily modified rendering the filtering system inaccurate.

The filtering systems process all packets in order to accept or drop them. So, they increase the response time and affect the system performance and availability [15]. Nevertheless, the efficiency of proactive approaches is better than that of the reactive approaches when they act with the DDoS attacks due to the fact that the malicious attacks are dropped before they access the targeted system.

Five existing mechanisms for handling the DDoS attacks will be presented in this section. This includes SOS, Kill-Bots, FOSEL, CLAD, and DaaS. CLAD will be provided in detail below while the remaining mechanisms will be described in a brief manner.

Cloud-Based Attack Defense System (CLAD):

The main objective of the CLAD is to secure the web servers through the provision of a strong security system in the shape of a network service which is usually operated on the vast structure of cloud (as a super computer) protecting from flooding attacks [17]. This super computer helps in overcoming the network layer attacks against any CLAD node which is a web proxy that is running on an application or virtual machine. The CLAD approach consists of a coalition of CLAD nodes and a DNS server where every CLAD node serves the function of a web proxy. It has diverse controlling initiatives that include admission control, congestion control initiatives, network layer filtering, authentication and pre-emption [17].

The concealment of the protected server from the public is quite relevant as the server may comprise of a sole server or a group of servers and merely allows the traffic which comes from the CLAD nodes to access the network. Furthermore, only CLAD nodes are aware of the IP address of the protected server so the DNS server response back to any received request from the internet with the IP address of a CLAD node.

A specific small file is fetched to exchange the healthy status of every CLAD node with its neighbors. The health status of the CLAD nodes are actually maintained by the authoritative DNS server that allocates the healthy CLAD nodes at the local DNS servers within a blink of an eye which makes the user in real time to choose the healthy CLAD node.

A session table holds active HTTP session keys where its optimum size can be determined by the present concurrent users. When the amount of created active HTTP session keys is decreased it refers to the admission control. A user can easily access the protected server by means of a CLAD system through a valid HTTP session key which is done for a specific time as the session key is saved in the cookie or attached with its URL. The other way of creating a session key is by hashing the user IP address and the expiration time utilized by a private hash function.

System of CLAD Works in the Following Way:

A client request is received by the DNS server that responses back with the IP address of a CLAD node which is chosen by determining its health status or load. Consequently, the client is verified by the CLAD node by means of a graphical turing test and afterwards allocates a session key further used to get the validation of the CLAD node if the user passes the test. It further transmits the request of the client to the protected web server [17].

The latency of CLAD is elevated due to all the packets of the clients must passing the components of the overlay system. The cloud infrastructure that serves the purpose of a network service, which safeguards the targeted web server, provides the web traffic access through it after ensuring that the request received is a HTTP request and drops the other traffics other than that. Moreover, the infrastructure of CLAD is only compatible for small enterprises.

More DDoS Countermeasures:

[18] proposed SOS as a reactive approach whereas the authors of [19] assert that SOS is known to be the first solution which utilized overlay techniques by which the target network indirect the received packets along with concealing the location of its protected web server to fight against the DoS attacks. In accordance with the view of [20] that SOS disallows the benign and anonymous users to access

the web servers and it also makes reaching to the protected server difficult for the users, it further protects the web server from getting an attack by providing a huge number of resources for which an attacker is required to formulate a successful attack. The major disadvantage of this technique is that the threats usually arise from spoofed IP addresses where they have the capability of launching DDoS attacks into any internal firewall. The limiting of response time in this technique is ignored in this case too.

Another reactive approach with the name of Kill-Bots is suggested by [21] that serve the function of a kernel extension for the protection of web servers from the application-layer DDoS attacks. It also makes use of the CAPTCHA for the verification of clients. Further, it alters the three-way handshake procedure linked to the TCP connection for safeguarding the verification approaches from flooding attacks as it does not establish a new socket until ending the TCP handshake procedure, as illustrated by [21]. Besides its advantages, it has certain drawbacks like it has increased the complexity as it enables the clients to test CAPTCHA many times, with the help of applying a bloom filter and admission control. At the same time, the risk is increased in this mechanism as it applies the protection approach on the server where it should be executed on the network edge which is the firewall. In this technique, the response time is elevated to a great extent due to the implementation of the prior mentioned factors.

Lastly, DaaS is a framework that is proposed by [11]. It establishes a metered pool comprising of resources that exceeding their counterpart in the botnets in order to simplify the process of controlling the idle resources which are free from usage. DaaS has more potency of elevating the response time as compared to prior mechanisms. Despite of all its merits, DaaS mechanism is not adequate to counteract the DDoS attacks by only utilizing the tool of puzzles due to their own drawbacks.

EDoS Countermeasures:

To protect the cloud from the EDoS attacks, there is a number of techniques have been proposed such as EDoS-Shield Framework, Shenai & Sandar approach, Enhanced EDoS-Shield Framework, and the In-Cloud eDDoS Mitigation Web Service. In the next subsections, there will be a brief description of the EDoS-Shield Framework and the In-Cloud eDDoS Mitigation Web Service (Scrubber Service) while the in-depth discussions are given in regard with the Enhanced EDoS-Shield Framework and the Shenai & Sandar approach.

1) Enhanced EDoS-Shield Framework

[22] developed a framework which is called the Enhanced EDoS-Shield to counteract the EDoS attacks which are generated by spoofed IP addresses. The key components are virtual firewalls (VF) and a cloud-based verifier node (V-Nodes). The firewall functions as a filter containing white and black lists that accumulate the IP addresses of the originating sources, Time to Live (TTL) values, a counter of unmatched Time to Live values in the black and white lists, and attack's initiating start time in the black list [21]. The framework developers utilized the Time-to-Live (TTL) value which is a field in the IP header to help identify the IP spoofed packet, i.e. packets developed from fake IP address. By making use of TTL, this method avoids declining a request from a user IP address that is placed on the blacklist. On the contrary, it examines the packet as it may be originated from a person who was a victim of an IP address spoofing attempt in the past. So, it stops DoS attacks on legitimate users if their IP addresses have been misused. In this method, the V-Node tests the first request from any source making use of graphic turing tests like CAPTCHA to bring up to date the lists according to the verification process outcomes.

The verification method will be applied in case the unmatched TTL counter does not go beyond the specified threshold. This will provide a new chance to the sources that have different TTL values to show their authenticity. The alteration of the TTL value between two definite ending points is restricted to a specified span of time by default. If the number of alterations goes above a specified threshold, then these alterations are believed to be as unusual and packets originating from the concerned IP address will be dropped without any further verification processes. The start time of the attack which is the moment of putting the origin of IP address in the blacklist is shown by the attack timestamp field. The objective of utilizing this field is to make the verification process at the V-Node undisclosed through the attack. For instance, if a packet appears during the attack's life-span with an origin of IP address that is present in the blacklist, it will be discarded without carrying a further verification method. On the contrary, in case the packet arrives after the attack's lifetime, a verification test will be performed given the probability that it may be a legitimate packet [21]. The drawback to this method is the rising in the latency as it examines every packet that comes at the firewall.

2) Sandar and Shenai Framework

[23] suggested a method that is dependent on a firewall performing the function of a filter. This system comprises a firewall and a client puzzle server. The user's request is received by the firewall which forwards it to the puzzle server. The user then gets a puzzle from the puzzle server to which he answers either correctly or wrongly. In case the user's reply is correct, the puzzle server will respond positively to the firewall. The firewall, after putting the user on the white list, will pass the request to the secured server to obtain the required services. On the contrary, if the firewall gets a negative response from the puzzle service, it will black list the user [23].

But this method has some shortcomings, particularly in dealing with the increased difficulty level of the puzzles that are sent to the legitimate users. Besides, this method has totally ignored the significance of limiting the response time although its inventors have evaluated the EDoS-Shield method, which was concerned with the solution of such latency [24].

More EDoS countermeasures:

[12] proposed the EDoS-Shield framework which utilizes CAPTCHA tests to determine whether the requests are originated from botnets or human users. It is the predecessor

of their Enhanced EDoS-Shield that cannot alleviate EDoS attacks initiating from spoofed IP addresses as it did not inspect the TTL values. Without efficiently shielding the target system, it only paid attention to limiting the response time.

[14] suggests yet another framework called In-Cloud eDDoS Mitigation Web Service (Scrubber Service). This service was developed as on demand service. Depending on the In-Cloud Scrubber Service, it creates and validates the puzzles at two dissimilar levels of difficulty to verify the clients in according with the nature of attack against the protected mechanism. There are two kinds of techniques termed as suspected mode and the normal mode. But the puzzles are the only focus of this method although they have their own drawbacks. Moreover, since under this method every packet needs to be verified, so it seems to be limited in its approach as the problem of response time, i.e. amount of time a message takes will still exist.

Existing Solutions Evaluation:

There is a need to use a comparative method to assess the performance of the mentioned DDoS and EDoS countermeasures that are evaluated above. The comparison is conducted based on validating the packets with a number of techniques, defending the scalability by reducing client's rate limiting, and decreasing the interval required to traverse on the system. The comparison process can be seen in Table 1.

It has been observed that the current methods paid attention on some factors and uncared for or are unsuccessful to come to terms with the needs of others. So, a new framework is designed by the author in a way that considers the above features in order to fill this gap.

3. The enhanced DDoS-MS Framework

The assessment of the current countermeasures indicates that the existing alleviating methods are not adequate

Table 1. Comparison between the Previous Frameworks' Performances.

The Framework	Strong Authentication	Protecting the Scalability	Decreasing the response time
SOS	☒	☒	☒
Fosel	☒	☒	✓
CLAD	☒	☒	☒
DaaS	☒	✓	☒
EDoS-Shield framework	☒	☒	✓
Sandar and Shenai	☒	✓	☒
Enhanced DDoS-Shield	☒	☒	☒
In-Cloud eDDoS Mitigation Web Service framework	☒	✓	✓

There is a need to propose a new solution that conducts strong verification of the origin of the traffic, protects the cloud scalability, and reducing the time required to traverse the system path. The proposed mechanism is created to perform these functions. This system knows about the earlier work; it comprises the key features of previous frameworks and overcomes their shortcomings. The novelty of this effort is offering a proactive defense of the cloud provider on their users' networks from the economic impact of the DDoS attacks by utilizing a new protection procedure, which meets the requirements of the above mentioned standards. Furthermore, for the verified clients, it will reduce the response time. This proposed system is termed as Enhanced DDoS-Mitigation System (Enhanced DDoS-MS). It is an improvement of the previous version (DDoS-MS) [24].

The design of this framework comprises five key components namely firewall, verifier node(s), client puzzle server, an Intrusion Prevention System (IPS) device, and a Reverse Proxy (RP) server in front of the shielded server(s).

The idea of the Enhanced DDOS-MS is to examine one packet sent by any origin by the verifier node(s), which utilizes the Graphical Turing Test (GTT) in validating the packets.

The precise job of the firewall is to filter the requests sent by the users. In the case of the requests are coming from illegitimate clients, the traffic will be stopped, and in the case of the legitimate users, the packets will be released through. Dependent upon the outcome of the verification procedure done by the verified node and monitoring methods conducted by the IPS and the RP, the firewall contains four lists for the origins of packets. These lists are white, black, suspicious, and malicious lists.

The IPS device is able to inspect the packets payloads to find out any maliscious software utlising Deep Packet Inspection (DPI) tools. The location of the protected servers is concealed by The Reverse Proxy (RP) server, which also control the load balance between these services and checks the rate of the packets flow with a view to finding out possible attempt of the DDoS attacks against these servies by establishing a pre-set threshold value for the number of packets from any user. Based on this pre-set threshold value, this attack-discovering procedure works in accordance with the number of packets at a particular time span.

With this proposed solution, only the first request will be checked by the verifier node while an IPS and an RP will handle the remaining requests. In case of suspected clients only, the puzzle server will be utilized in this solution to control them when they are going beyond the threshold value in the reverse proxy.

In case any malicious software is found in the packet by the IPS, its IP address is put on the Malicious List. Reverse Proxy (RP) performs the last level of the monitoring procedure. Suspicious clients who make attempts to devastate the system by sending an overwhelming number of requests that can pass the earlier monitoring layers are detected by the Reverse proxy. In this way, the origin of such suspicious users will be put in the Suspicious List.

If the firewall receives any request from a suspicious user, it will send it onward to the client puzzle server which forwards

a crypto puzzle to that suspicious user with a view to delaying this by taking a particular duration and computational resources on his side with a view to defend the system from the threat of DDoS attacks. Unlike its usage in the DDoS-MS framework, the puzzles in this enhanced solution are utilized as a reactive measure that is only against the suspicious clients.

As a result, the legitimate client will not be forced to undertake further tests after successfully going through the validation procedure. Until his legitimacy is suspected due to going beyond the threshold value of the traffic, or sending packets contains malicious software, or altering the TTL values of the packets, he/she will not be checked in the application layer using a GTT or in the network layer by the crypto puzzles.

This three-stage mechanism is aimed enabling each part to perform a particular function as it equally allocates the monitoring responsibilities among these three layers. The Enhanced DDoS-MS framework's design has been illustrated in Fig. 1:

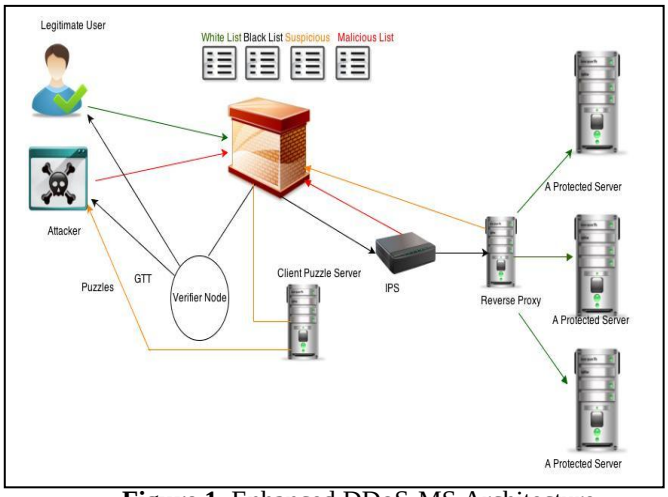

Figure 1. Enhanced DDoS-MS Architecture

The suggested system is founded on the following assumptions with a view to reducing its scope:

1. The customer's system must use this framework which can also be utilized in the provider's system.
2. The adversary's objective is to conduct DDoS attacks against the cloud to disturb its pay-per-use model by taking advantage of the vulnerabilities in the clients' authentication system.
3. The framework examines one packet which its gets from any origin, supposing that sources' IP addresses are constant and the packets are not fragmented, so the TTL values will not be altered according to the various paths the packets can utilize to reach to the target.

The objective of testing only one packet and then monitoring the remaining packets is to improve the performance of the EDoS-Shield framework in the reduction of the response time. The function of the verifier node is to test the sources and differentiate between the legitimate user and the bots.

What distinguishes the Enhanced DDoS-MS from other frameworks is its capability to pay attention to all three challenges at the same time, protecting the cloud from DDoS attacks, which involves tough verification procedure, defends the scalability advantage of the cloud and ensures reduction in the response time.

The following scenarios are designed to explain the Enhanced DDoS-MS mechanism:

A. The Scenario of testing the first packet

1. A client forwards a request to the shielded server.
2. The request is received by the firewall which checks its lists.
3. The firewall forwards the request to the verifier node in case the packet source's address is not present on either list.
4. A GTT test is sent to the client by the verifier node.
5. The verifier node sends a positive acknowledgement to the firewall in case the client emerges successful in the test. In case of failure, a negative response is received by the firewall.
6. If the verifier nod forwards a negative outcome to the firewall, it will refuse the request. As a result, the client's IP address, its TTL (Time To Live) value, and the original time of the attack (timestamp) will be added to the black list.
7. Otherwise, the client's IP address and its TTL value will be placed in the white list.
8. The request will be sent to the IPS device and the RP server respectively by the firewall down to the protected server.
9. Last but not the least, the required service will be provided directly to the client.

B. If the source of the subsequent packets is a legitimate user (On the White List)

1. The firewall checks its lists as soon as it receives the packets.
2. The white list contains the packet source's address. If the packet's TTL value is matching to the Time to Live value registered on the white list, then the firewall, through the IPS and the RP, will pass the packet on to the protected server.
3. Otherwise, the request will be passed to the verifier node for conducting the GTT test. This measure will find out whether the IP address recorded on the white list is a victim of a spoofing attack; and stop the attacker from exploiting the white list addresses.
4. In case a negative result is forwarded to the firewall from the verifier node, the flow will be stopped and the client's details will be excluded from white list.
5. Otherwise, this request is sent by the firewall with all subsequent requests from this client (in case their Time to Live values are similar to the recorded TTL values existing on the white list) to the protected server and the details of this user are updated on the white list.
6. Then these packets pass through the IPS, which inspecting them and in case it find out malicious software stuff, then it stops it from proceeding further and also brings it to the notice of the firewall.
7. The malicious IP address will be moved from the white list (WL) to the malicious list (ML) by the firewall.
8. Otherwise, the flow will continue to proceed through the RP. If the RP detects that the number of requests is more than pre-set threshold value, it will stop the suspicious packets and bring it to the notice of the firewall.
9. The firewall will transfer the suspicious IP address from the white list (WL) to the suspicious list (SL).

10. Otherwise, the requests will be passed to the protected server and their source will get his requested service.

C. If the source of the subsequent packets is on the Black List

1. In this scenario, the source address of the requests lies on the black list; the firewall evaluates the recorded values of the source Time To Live, and the start time of the attack. If the TTL value is in accordance with the registered TTL value as existing on the black list, or if the initiation time of the request has the same start time as the previous malicious request, then the firewall will drop the current packet and update the adversary's details on the black list.
2. Otherwise, the firewall will forward the packet to the verifier node to verify it utilising a Graphical Turing Test (GTT). This measure gives the victim of a past spoofing attack an opportunity to confirm his authenticity.
3. In case a negative outcome is received by the firewall from the verifier node, then the attacker's details will be bring up to date on the black list and this client's present packet will be dropped.
4. Otherwise, the request will be forwarded to the protected server to obtain the desired services, and the client's source IP address will be excluded from the blacklist.
5. The requests will then go through the IPS, which will stop it from proceeding in case it finds out malicious software contents, and will bring it to the notice of the firewall, which will add this malicious IP address to the malicious list.
6. Otherwise, the traffic will pass through the RP. If the RP detects that the number of packets is beyond its pre-set threshold value, then it will reject it and report it to the firewall.
7. The suspected malicious user's IP address will be put on the suspicious list (SL) by the firewall.
8. Otherwise, the flow will be sent to the request server, and the needed-service and information will be sent directly to the client.

D.If the source of the subsequent packets is a suspicious user (On the Suspicious List)

1. In this situation, the source address of the suspected client is present on suspicious list; the packet from this source is forwarded to the puzzle server by the firewall. The puzzle server subjects the user to hard crypto puzzle checking.
2. In case the client passes this process, the puzzle server forwards positive result to the firewall. If the client fails in this test, a negative affirmation will be sent to the firewall.
3. The existing packet will be dropped in case the firewall receives a negative acknowledgement from the client puzzle server, and the client's IP address will be included in the malicious list. In case of positive outcome, the firewall will forward the requests to the IPS for checking purposes.
4. In case the IPS finds out malicious software substance in the packets, then it will exclude it and also report it to the firewall.
5. As a result, the IP address of this sender will be moved by the Firewall to malicious list from the suspicious list.
6. Otherwise, the traffic will proceed through the RP checking. If the RP finds that the number of packets is beyond the pre-set threshold value, it will stop it from proceeding and will report it to the firewall.
7. This suspicious address will be moved from suspicious list to the malicious list by the firewall.
8. Otherwise, the flow will be passed to the requested server, and the required service and information will be forwarded directly to the client.

E.If the source of the subsequent packets is a malicious user (On the Malicious List)

In this situation, the request's source's address is included in the malicious list due to clients past packets which were full of viruses or contained worm, or its source address was found to be involved in attempts to conduct DDoS attacks against the network and subsequently continuing with te same attempt to overwhelm the network, or not passing the puzzle checking. As a result, all incoming requests from this client are rejected as well as the access to the network is out rightly denied by the firewall.

Enhanced DDoS-MS Evaluation:

Laboratory environment, where the implementation of Enhanced DDoS-MS was tested, consists of three main domains as shown in Fig.2. The first one is the outside domain that represents area of not malicious end users as well as malicious attackers. Second one is the decision making domain that refers to a group of techniques that are used to verify the legitimacy of users such as the firewall, verification node, puzzle server, IPS, and the Reverse proxy. The last one is the protected area that incorporates protected servers and services.

Multiple testing scenarios were used to prove the concept of the Enhanced DDoS-MS. The actual set up was based on the generating the traffic, from the outside domain, though decisions making domain into the protected domain.

The outside domain was used as a source of traffic that sent to a web server in the protected area. This simple request was captured and recorded with low level network analyzer Wireshark. After that, this data were exported and used in traffic generator PackETH, where is a possibility to change the parameters of the particular packets and amplification the volume of the traffic. Wireshark was running on each interface to monitor the traffic in the testbed. The Protected domain in the test bed presents the area of the protected server.

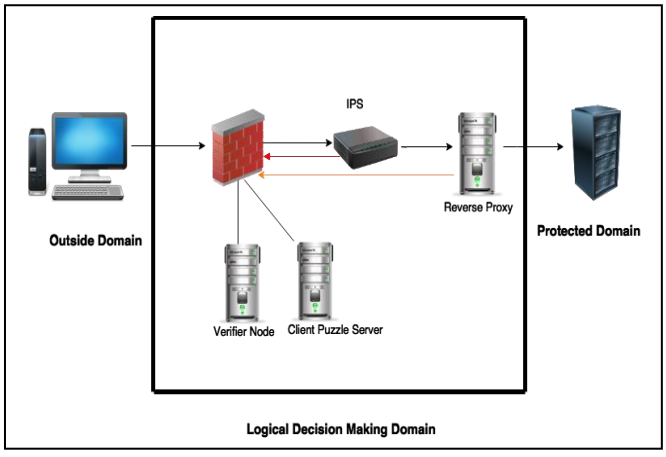

Figure 2. Implementation environment for a testbed of Enhanced DDoS-MS

The Decision making domain included above mentioned techniques, these techniques were implemented and simulated by algorithm written in C++ programming language. For a low level packet handling, pcap and libnet library were used. Firewall fully implemented the decision process as suggested in Enhanced DDoS-MS framework. It is a command line-based program that has the possibility of showing internal actions in a terminal with setting up an appropriate program attribute. The internal actions of the firewall can be seen in Fig. 3.

```
Enhanced DDoS-MS Firewall: Started.
Outside Interface: eth1
Protected areal Interface: eth0
----------------------
Packet recieved: IP src: 10.0.0.1 IP dest: 192.168.0.4
Verification node: Legitimate source: IP - 10.0.0.1, TTL - 64.


Packet recieved: IP src: 10.0.0.1 IP dest: 192.168.0.4
List match found: src: 10.0.0.1
WL: Source IP - 10.0.0.1, TTL - 64  found. Packet goes through.
IPS: This trafic is not malicious. Packet goes through.
RP: Load in normal. Packet goes through.
Packet went through the firewall.

Packet recieved: IP src: 10.0.0.1 IP dest: 192.168.0.4
List match found: src: 10.0.0.1
WL: Source IP - 10.0.0.1, TTL - 64  found. Packet goes through.
IPS: This trafic is not malicious. Packet goes through.
RP: Load in normal. Packet goes through.
Packet went through the firewall.


Packet recieved: IP src: 10.0.0.1 IP dest: 192.168.0.4
List match found: src: 10.0.0.1
WL: Source IP - 10.0.0.1, TTL - 64  found. Packet goes through.
IPS: This trafic is not malicious. Packet goes through.
RP: Load in normal. Packet goes through.
Packet went through the firewall.
```

Figure 3. Firewall Actions

4. Results

In the implementation, there are two variables that will be compared. They are the traffic load and the traffic intensity. Thus, two experiments will be conducted; the first one involves changing the load between 500 to 5000 packets and fixing the intensity to be constant. In the second experiment, the load will be fixed to be constant at 4000 ICMP packets and the intensity will be changed by dividing the whole stream into four sub-streams and two different orders.

The purpose of conducting the implementation in the above suggested way is to examine the influence of changing the load and the intensity on the capability of the proposed solution to handle the received packets at an acceptable level of response time beside providing the required security.

The two experiments are described below and the main finding which is a comparison between the values of the average response time of the conducted scenarios will be summarized at the end of this section in order to prove the effectiveness of the proposed framework in decreasing the latency for the legitimate users regardless the change in the load or intensity of the received traffic.

The First Experiment:

500 ICMP packets were generated as a typical load situation and sent through the proposed framework to the protected domain. Fig. 4 shows the variance of the response time for 500 ICMP packets.

It is clear that the majority of the packets are completely served between 0.3 and 0.5 ms. The maximum response time is about 0.857 ms while the minimum is 0.301 ms. So, the average is 0.419 ms. This scenario is conducted as a base line of the next experimental scenarios.

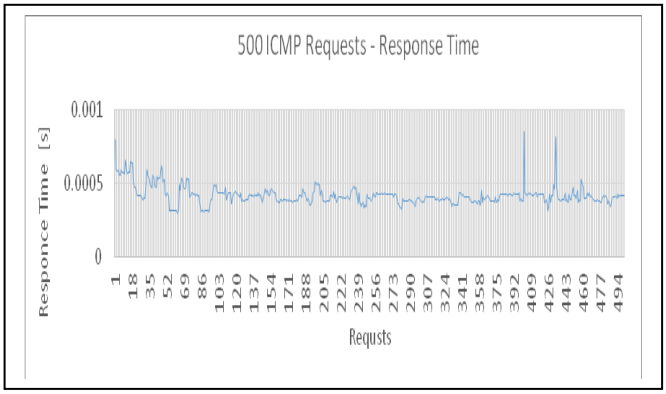

Figure 4. The Response Time Variance of 500 ICMP Packets

To evaluate the effect of increasing the number of packets on the response time, the previous amount i.e., 500 packets is multiplied by 10 to be 5000 ICMP packets. Fig. 5 shows that the maximum response time is 1.014 ms for the packet 4639 while the minimum is 0.304 ms in the packet 656. Therefore, the average is 0.426 ms.

That means the response time is almost constant for the whole amount of the tested ICMP packets either if the sample is 500 or 5000 packets because the actual difference in the average response time is 0.007 ms (7µs). The increasing number of requests did not affect the response time. Thus, the constant average response time is a good feature of the protection system as it is not get to be overwhelmed by the higher streams. So, it can be resilient under the attacks.

That means the response time is almost constant for the whole amount of the tested ICMP packets either if the sample is 500 or 5000 packets because the actual difference in the average response time is 0.007 ms (7µs). The increasing number of requests did not affect the response time. Thus, the constant average response time is a good feature of the protection system as it is not get to be overwhelmed by the higher streams. So, it can be resilient under the attacks.

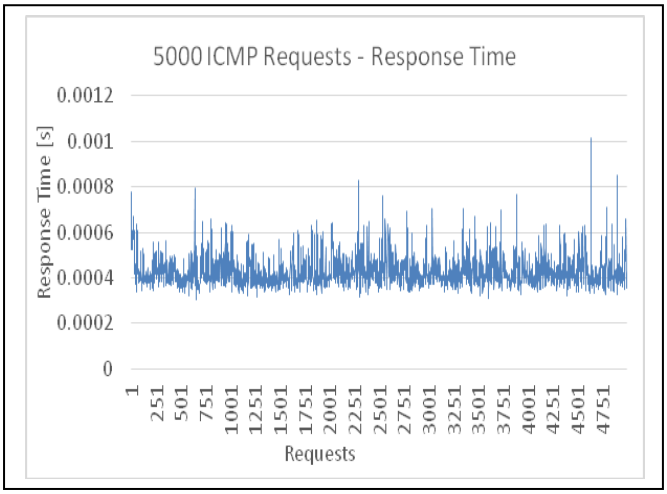

Figure 5. The Response Time Variance of 5000 ICMP Packets

The Second Experiment

Now, three scenarios will be conducted experimentally in order to evaluate the proposed framework in terms of limiting the response time when the received load is divided into four streams with different traffic intensities. Each stream has the same number of ICMP packets but the traffic intensity is different. Thus, the whole load of 4000 packets will be sent through the framework in burst intensities i.e., diverse transfer rate and in three different orders (scenarios).

For comparison purposes, 4000 packets as one stream in a constant traffic intensity are sent through the firewall. Fig. 6 shows that the maximum response time is 0.829 ms while the minimum is 0.304 ms. Therefore, the average is 0.426 ms.

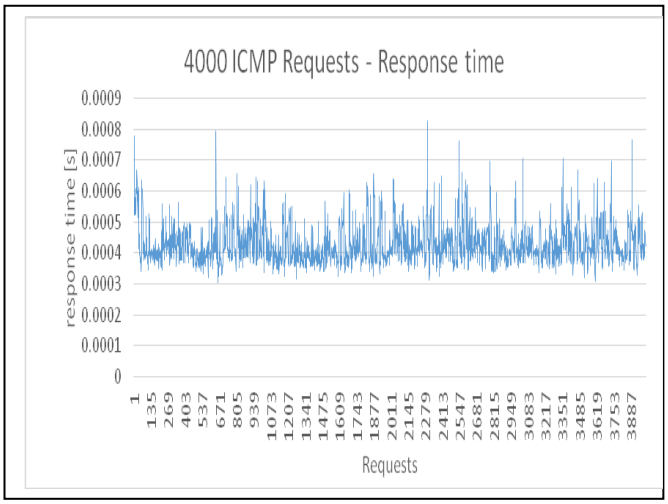

Figure 6. The Response Time Variance of 4000 ICMP Packets

Actually, the majority of packets are served in a range of 0.1 ms from the average level as shown in Table 2. It is clear from the table that 93.49 percent of all packets are handled within 0.1 ms time window from the average response time.

Table 2. Distribution of Response Time Values of 4000 ICMP Packets

Time Difference with regard to the Average Response Time [ms]	Number of Packets [Responses]	Percentage [%]
1	252	6.30%
0.1	443	11.08%
0.05	817	20.43%
-0.05	1870	46.75%
-0.1	609	15.23%
-1	9	0.23%
Sum:	4000	100%

After completing the following experiment, the average response time of each scenario whole load can be compared with its counterpart in the previous result in Fig. 6 that apply constant intensity and with the other different intensities scenarios results. Thus, the two scenarios results are presented and analyzed in the following part:

1. First Scenario

This scenario has four streams. Every stream consists of 4000 requests and 4000 responses so the total number of packets is 8000 packets. The traffic intensities of the streams are diverse between 50 packets per second (pps) to 1000 pps in the order (100 pps, 50 pps, 500 pps, 1000 pps). The variance of these streams is shown in Fig. 7.

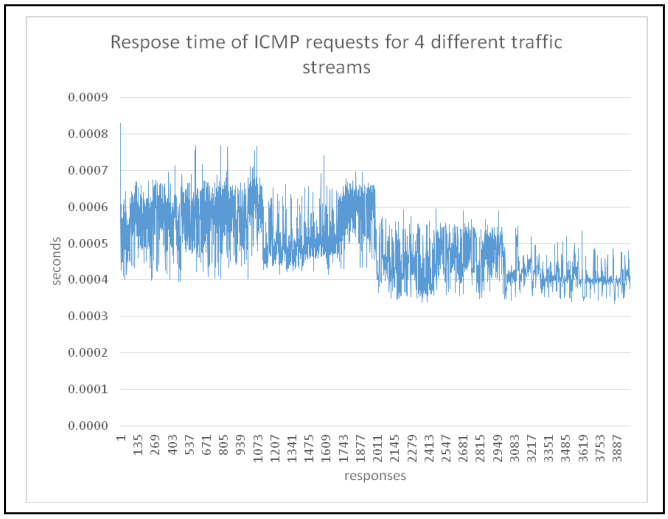

Figure 7. The Response Time Variance of the First Scenario Streams

To facilitate analyzing the results on Fig. 7, Table 3 presents the average, minimum and maximum response time of each stream as well as for the whole load. It is noticeable that highest intensity which is 1000 pps achieves the lowest average response time that is 0.41 ms. The difference between the average response time of the highest intensity and the lowest one is 0.13 ms (130 µs). The average response time for the whole load of streams in the current order is 0.49 ms.

Table 3. Summary of the First Scenario Streams Response Times

Stream number	Average response time [ms]	Minimum response time [ms]	Maximum response time [ms]
Stream # 1	0.56510	0.39400	0.83000
Stream # 2	0.54449	0.41500	0.76700
Stream # 3	0.45389	0.33700	0.59500
Stream # 4	0.41035	0.33300	0.54900
Whole load	0.49347	0.33300	0.83000

2. Second Scenario

After presenting the above results of the first scenario, the second scenario shows how the change of the streams order can affect the framework performance. In this scenario, the streams will be sent in a different order. The streams order is rearranged to be 500, 1000, 50, and 100 pps respectively as stated in Table 4.

Table 4. The Second Scenario Streams Description

Stream number	Number of ICMP requests send within the stream	Intensity of packet generation per second within the stream	Total number of packets
Stream # 1	1000	500	2000
Stream # 2	1000	1000	2000
Stream # 3	1000	50	2000
Stream # 4	1000	100	2000
SUM	4000	-	8000

Fig. 8 that shows the variance of the streams. It is clear from the figure that the second stream which has the intensity of 1000 pps achieves the lowest average response time although it is sent as a second stream this time not the fourth as in the previous scenario.

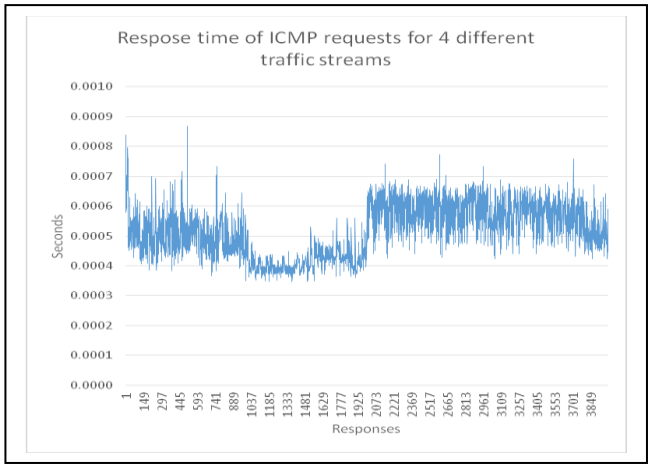

Figure 8. The Response Time Variance of the Second Scenario Streams

The full explanation of the streams response times is presented in Table 5

Table 5. Summary of the Second Scenario Streams Response Times

Stream number	Average response time [ms]	Minimum response time [ms]	Maximum response time [ms]
Stream # 1	0.50145	0.34900	0.86800
Stream # 2	0.41440	0.34700	0.56100
Stream # 3	0.58858	0.42500	0.77300
Stream # 4	0.55691	0.42300	0.75800
Whole load	0.51535	0.34700	0.86800

It is observable from the table that the lowest intensity (50 pps) which is represented by the third stream achieves the highest average response time (0.59 ms). 170 µs is the difference between the average response time of the second and third streams that represent the highest and lowest intensity respectively. The whole load of streams with this order has an average response time value of 0.52 ms. This proves that changing the streams order did not affect the average value of the response time.

The overall findings are presented in Table 6 that compares the averages of response time of all streams. The average value is 0.48 ms for either a single stream with constant intensity or a load of streams (burst intensities). It means that the proposed framework can perform the protection function in a limited time window of approximately 0.05 ms (50 µs) from the average response time despite the variations of packets numbers and streams intensities. This reflects the effectiveness of the solution in achieving the required objective which is providing the security to the protected system besides limiting the response time for the legitimate users despite the diversity of the order of the received streams.

Table 6. A Comparison of Average Response Time Values of the whole conducted streams

Stream	Average Response Time [ms]
4000 Packets [1 stream]	0.42
The First Scenario [4 streams]	0.49
The Second Scenario [4 streams]	0.52
Total Average value:	0.48

These averages are reasonable according to the different techniques that are implemented in the proposed system. The effectiveness of the Enhanced DDoS-MS framework lies on embracing the potential malicious and suspicious requests in its work flow without affecting the legitimate users neither in their ability to access the system at any time nor in the delay that has been added as a result of implementing a strong protection technique that ensures the availability of the services and protecting from the DDoS attacks and EDoS attacks. Moreover, the results prove the framework's scalability under varied loads and traffic intensities with different orders.

5. Open Research Issues

Additional work in this project requires framework enhancement and comprises various conditions that were not studied in this research. For instance:

1. Enhancing the current framework to include the case of making use of dynamic IP addresses.
2. Involving the status of IP packet fragmentation.
3. Selecting further packets at random for more tests.
4. Safeguarding the cloud user's network that permits BYOD trend within its internal system.

Furthermore, the performance of the proposed solution can be further enhanced to render it more effective against genuine attacks. Adversaries are adopting improved methods to make their attempts of sabotage more successful. For that reason, upcoming research will focus on evaluating the proposed framework in highly developed testing scenarios, which may comprise multifarious complex legitimate clients action scenarios and various complicated malicious attack behavior situations:

A. Complicated legitimate user behavior scenarios

The client is a human being, who can err or just act not in usual way. At times, his/her attempts seem to be malicious ones despite the fact that he/she does not want to cause

damage to the server:

1. The instance of CAPTCHA test failure because of the keyboard issues or inadequate skills of the legitimate clients. The said clients will be denied access to the server despite the fact that they are not attackers but the existing design of the solution removes them from the white list.
2. Improving the solution to take in its consideration the flash crowd phenomenon. In such a scenario, the legitimate clients successfully pass the Turing test and the other checks but they flood the protected server by a large number of requests from a large number of legitimate sources.

B. Complex malicious attacks behavior scenarios

Three different phases like locations, layers, and behavioral modifications are covered by these scenarios. Looking at the layers, the authors of this mechanism aim at utilizing attacking methods on multiple ISO/OSI layers:

1. Safeguarding the cloud from the intricate attacks that initiated by taking advantage of the related shortcomings in multiple layers of the network targeted by the attackers. Attaining such effectiveness makes the solution more effective and provides strong defense to the cloud.
2. Distributing the attack source's locations through wider geographical ranges to emulate the persistent adversaries who targets to damage particular network.
3. Modifying the attackers' actions by exchanging the attacks recurrently between various groups of attackers at random intervals. This method renders the finding out of attack sources very hard. Most significantly, it renders the attack very difficult to be identified by the security measures that applied in the target's side.

6. Conclusion

Cloud computing has become an essential backbone infrastructure for many businesses and industries. Therefore, it is becoming a very important subject for security threats.

The paper introduces an efficient method called Enhanced DDoS-MS that aims at protecting cloud resources against one of the key types of security threats, namely DDoS and EDoS attacks.

The proposed method has been evaluated through a real setup which showed that it outperforms existing methods in efficiency and low delay caused by its verification stages. It limits the average response time despite the variations of packets numbers and streams intensities. This reflects the effectiveness of the solution in achieving the required objective which is providing the security to the protected system besides limiting the response time for the legitimate users despite the diversity of the loads and traffic intensities with different orders.

References

[1] Intel, "What's Holding Back the Cloud?," 2012. [Online]. Available: http://www.intel.com/content/www/us/en/cloud-computing/whats-holding-back-the-cloud-peer-research-report.html. [Accessed: 26-Sep-2014].

[2] W. Alosaimi and K. Al-Begain, "An Enhanced Economical Denial of Sustainability Mitigation System for the Cloud," in 2013 Seventh International Conference on Next Generation Mobile Apps, Services and Technologies, pp. 19–25,2013.

[3] K. Matthew and A. Md,"An Effective Way of Evaluating Trust in Inter-cloud Computing". International Journal of Computer Network and Information Security (IJCNIS), Vol. 9, No. 2, pp.36-42, 2017.

[4] ENISA, "Cloud Computing Benefits, Risks and Recommendations for Information Security," Eur. Netw. Inf. Secur. Agency, pp. 9–10, 2009.

[5] L. Kavisankar, C. Chellappan, S. Venkatesan and P. Sivasankar, "Efficient SYN spoofing Detection and Mitigation Scheme for DDoS attack", Second International Conference on Recent Trends and Challenges in Computational Models (ICRTCCM), pp. 269 – 274, 2017.

[6] Y. Wang, L. Liu, C. Si and B. Sun, 'A novel approach for countering application layer DDoS attacks", IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), pp.1814 – 1817, 2017.

[7] A. Bhardwaj, G. Subrahmanyam, V. Avasthi, H. Sastry and S. Goundar, "DDoS Attacks, New DDoS Taxonomy and mitigation solutions — A survey", International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES), pp. 793 – 798, 2016.

[8] A. Khare, J. Rana and R. Jain, "Detection of Wormhole, Blackhole and DDOS Attack in MANET using Trust Estimation under Fuzzy Logic Methodology", International Journal of Computer Network and Information Security (IJCNIS). Vol. 9, No. 7, pp.29-35, 2017.

[9] Y. Wang, L. Liu, C. Si and B. Sun, "A novel approach for countering application layer DDoS attacks", IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), pp. 1814 – 1817, 2017.

[10] C. Hoff, "Cloud Computing Security: From DDoS (Distributed Denial Of Service) to EDoS (Economic Denial of Sustainability)," Rational Survivability, 2008. [Online]. Available: http://rationalsecurity.typepad.com/blog/2008/11/cloud-computing-security-from-ddos-distributed-denial-of-service-to-edos-economic-denial-of-sustaina.html. [Accessed: 27-Sep-2012].

[11] Khor and A. Nakao, "DaaS: DDoS Mitigation-as-a-Service," in 2011 IEEE/IPSJ International Symposium on Applications and the Internet, pp. 160–171,2011.

[12] M. Sqalli, F. Al-Haidari, and K. Salah, "EDoS-Shield - A Two-Steps Mitigation Technique against EDoS Attacks in Cloud Computing," in 2011 Fourth IEEE International Conference on Utility and Cloud Computing, pp. 49–56,2011.

[13] C. Hoff, "A Couple Of Follow-Ups On The EDoS (Economic Denial Of Sustainability) Concept," Rational Survivability, 2009. [Online]. Available: http://www.rationalsurvivability.com/blog/2009/01/a-couple-of-follow-ups-on-the-edos-economic-denial-of-sustainability-concept/. [Accessed: 26-Jan-2013].

[14] M. Kumar, P. Sujatha, V. Kalva, R. Nagori, and A. Katukojwala, "Mitigating Economic Denial of Sustainability (EDoS) in Cloud Computing Using In-cloud Scrubber Service," in 2012 Fourth International Conference on Computational Intelligence and Communication Networks, pp. 535–539,2012.

[15] P. Mell and T. Grance, "The NIST Definition of Cloud Computing. National Institute of Standards and Technology," Vol. 53, No. 6, p. 50, 2009.

[16] H. Beitollahi and G. Deconinck, "FOSeL: Filtering by Helping an Overlay Security Layer to Mitigate DoS Attacks," 2008 Seventh IEEE Int. Symp. Netw. Comput. Appl., pp. 19–28, 2008.

[17] Beitollahi and G. Deconinck, "Analyzing well-known countermeasures against distributed denial of service attacks," Comput. Commun., vol. 35, no. 11, pp. 1312–1332, 2012.
[18] W. Morein, A. Stavrou, D. Cook, A. Keromytis, V. Misra, and D. Rubenstein, "Using graphic turing tests to counter automated DDoS attacks against web servers," in the 10th ACM conference on Computer and communication security - CCS '03, pp. 8–19,2003.
[19] P. Du and A. Nakao, "DDoS defense as a network service," 2010 IEEE Netw. Oper. Manag. Symp. - NOMS 2010, pp. 894–897, 2010.
[20] N. Kumar and S. Sharma, "Study of intrusion detection system for DDoS attacks in cloud computing," in 2013 Tenth International Conference on Wireless and Optical Communications Networks (WOCN), pp. 1–5, 2013
[21] S. Kandula, D. Katabi, M. Jacob, and A. Berger, "Botz-4-sale: Surviving organized DDoS attacks that mimic flash crowds," Proc. 2nd Symp. Networked Syst. Des. Implement., pp. 287–300, 2005.
[22] F. Al-Haidari, M. Sqalli, and K. Salah, "Enhanced EDoS-Shield for Mitigating EDoS Attacks Originating from Spoofed IP Addresses," in the 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications, pp. 1167–1174, 2012.
[23] V. Sandar and S. Shenai, "Economic Denial of Sustainability (EDoS) in Cloud Services using HTTP and XML based DDoS Attacks," Int. J. Comput. Appl., vol. 41, no. 20, pp. 11–16, Mar. 2012.
[24] W. Alosaimi and K. Al-Begain, "A New Method to Mitigate the Impacts of Economical Denial of Sustainability Attacks Against the Cloud," in The 14th Annual Post Graduate Symposium on the Convergence of Telecommunications, Networking and Broadcating, pp. 116–121,2013.