EDoS-ADS: An Enhanced Mitigation Technique Against Economic Denial of Sustainability (EDoS) Attacks

Ahmad Shawahna, Marwan Abu-Amara, Ashraf S. H. Mahmoud, and Yahya Osais

Abstract—Cloud computing is an essential technology for the future of the Information Technology (IT) industry. However, the cloud security level is identified as the biggest challenge facing cloud providers and a major concern for cloud adopters. Economic Denial of Sustainability (EDoS) attack is one of the major threats targeting the cloud. The EDoS attack exploits the cloud elasticity and auto scaling features to charge a cloud adopter bill an excessive amount of cost leading to large-scale service withdrawal or bankruptcy. A novel reactive approach referred to as the EDoS Attack Defense Shell (EDoS-ADS) is proposed to mitigate EDoS attacks while taking into account most of the existing mitigation techniques drawbacks. Specifically, the EDoS-ADS has the ability to identify the legitimacy of clients even if they belong to a Network Address Translation (NAT) based network. Thus, EDoS-ADS is the first known technique that effectively prevents an EDoS attack from blocking an entire NAT-based network from accessing the cloud. The EDoS-ADS effectiveness in terms of response time, CPU utilization, throughput, and cost is evaluated using a CloudSim simulator. The simulation results show that EDoS-ADS outperforms other mitigation techniques, and successfully differentiates between legitimate and attacker clients even when they belong to the same NAT-based network.

Index Terms—Cloud security, Economic Denial of Sustainability (EDoS) attack, CloudSim simulator, network address translation.

1 Introduction

Cloud computing presents a model in which on-demand network access to the computing resources, utility-based pricing model, and dynamic resource assignment is granted as a service over the Internet [1]. The cloud resources can be rapidly provisioned and freed with minimal cloud provider intervention by using an auto scaling feature. The feature automatically triggers a scale up that allocates more resources to handle the high load, and a scale down that releases these resources when the load returns to normal. The auto scaling feature is activated by monitoring parameters such as CPU utilization, memory usage, response time, and network bandwidth.

Based on a survey conducted by the International Data Corporation (IDC) [2], security is cited as the major challenge for the cloud computing model. Nearly 87% of the Information Technology (IT) executives reported the cloud security as the principal challenge prohibiting its adoption. With security being a top concern that hinders cloud computing environments [3], it became a major field of study. However, the aspect of cloud computing security is wide and general. Hence, it is vital to consider two well-known types of network security threats; Denial of Service (DoS) and Distributed Denial of Service (DDoS). The DoS and DDoS attacks overwhelm a network infrastructure by employing a number of infected machines to perform unwanted operations intended to cause damage to the network infrastructure [4]. Such attacks make, for example, an organization website unavailable to end users due to an exhaustion of its resources.

• *The authors are with the Department of Computer Engineering, King Fahd University of Petroleum and Minerals, Dhahran, KSA, 31261. E-mail: { g201206920, marwan, ashraf, yosais }@kfupm.edu.sa*

As stated earlier, the cloud computing model permits the customers to scale their resources in size and availability. Note that the cloud customers are charged depending on the pay as you go premise of the cloud's resources. Such a service model may appear to overcome the effects of a DDoS attack where the resource bottlenecks are eliminated. However, this model merely transforms the DDoS attack to a new strain of attacks that targets the cloud adopters' economic resource to charge their bill an excessive amount of cost. Such an attack is originally labeled as Economic Denial of Sustainability (EDoS) attack [5]. Unlike a DDoS attack that can prevent legitimate users from accessing the service for a certain amount of time, an EDoS attack can prevent a cloud adopter from delivering the service forever if the attack leads to bankruptcy.

Wang et al. [6] conducted several EDoS attacks experiments to demonstrate how severe the attack is to the cloud consumers. The authors achieved the EDoS attack by abusing Google, Microsoft, LinkedIn, and other public third-party services easily and at a very low cost. Therefore, a clear consideration of common attack mitigation strategies is highly needed to address the consequences of EDoS attacks as discussed in [4], [7], [8]. Since DDoS attacks in the cloud transform into EDoS attacks, the access management to the cloud services is ranked as the top most significant DDoS attack mitigation strategy [9]. Thus, an efficient EDoS attack mitigation technique should effectively control the access to the cloud resources.

Hence, in this paper, we review the major existing EDoS mitigation techniques and state their drawbacks. Then, a new EDoS mitigation approach referred to as the EDoS Attack Defense Shell (EDoS-ADS) is proposed. The EDoS-ADS approach avoids false negatives while allowing the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2018.2805907, IEEE Transactions on Cloud Computing

legitimate clients access to the cloud services.

The remainder of the paper is organized as follows. Section 2 provides a literature review of the major EDoS attack mitigation techniques, and the methods used to identify clients that belong to a Network Address Translation (NAT) based network. Section 3 presents the proposed EDoS-ADS technique. Section 4 discusses the simulator implementation used to evaluate the EDoS-ADS. Section 5 presents the simulation results and analysis. Finally, we conclude and describe the future directions of this paper in section 6.

2 Literature Review

Many researchers have focused on the EDoS attack due to its severe impact on the cloud adopter's bill. Hence, a literature review of the major EDoS attack mitigation techniques is presented. Moreover, the methods used to distinguish between clients that belong to a NAT-based network are discussed. Discussing such methods is important as any practical EDoS mitigation technique must be able to do so for the purpose of preventing the unintentional blocking of an entire NAT-based network. Note that an example of a NAT-based network is an enterprise network that subscribes to the cloud and uses a NAT gateway router.

2.1 EDoS Attack Mitigation Techniques

To illustrate the need for specific EDoS attack mitigation techniques we first review existing DDoS attack mitigation techniques and their suitability to address EDoS attacks. Consequently, Osanaiye et al. [4] reviewed 96 approaches related to DDoS attacks and defenses in the cloud computing environment. The authors categorized DDoS mitigation techniques into three groups; anomaly based, signature based, and hybrid. Examples of the anomaly based techniques include MTF [10] and E-EMD [11]. On the other hand, examples of the signature based techniques include the filter tree approach [12], attack pattern detection scheme [13], and ensemble-based multi filter feature selection method [14]. Osanaiye et al. [4] concluded that the anomaly based detection techniques can result in a high misclassification rate and a high resource consumption. Similarly, the authors remarked that, in general, signature based DDoS defenses may result in a high rate of false negatives due to their failure to detect zero-day attacks. Similar reviews to [4] are provided by [15], [16], and [17].

The aforementioned reviews highlighted that most DDoS mitigation techniques are designed for a specific type of DDoS attack. As such, a more general DDoS attack can transform the DDoS attack to an EDoS attack. Thus, the majority of the traditional DDoS detection and mitigation techniques are not suitable to detect and mitigate DDoS attacks that result in EDoS attacks [18]. Furthermore, these reviews as well as [19] pointed out an important finding which is that the request Inter-Arrival Time (IAT) can be greatly helpful in detecting different forms of DDoS attacks including those that result in EDoS attacks. It is worth noting that the IAT observation is utilized in our proposed EDoS-ADS to mitigate EDoS attacks. Finally, Bhingarkar et al. [18] highlighted that an EDoS attack typically spans a long period of time unlike a short-lived DDoS attack.

Based on the previous discussion, dedicated EDoS attacks detection and mitigation techniques are clearly needed. Thus, we present next the major existing EDoS mitigation techniques while stating their main shortcomings.

Bhingarkar et al. [18] and Somasundaram [20] provided surveys of EDoS attack detection and mitigation techniques. The two surveys show that the existing EDoS attack signature-based detection techniques suffer from high false positives when the EDoS attack is the result of an unknown DDoS attack. In addition, the two surveys stated that many of the existing EDoS attack mitigation techniques have many drawbacks including adding a huge overhead on the cloud system as well as the possibility of blocking an entire NAT-based network. The two surveys conclude by highlighting that detecting and mitigating EDoS attacks effectively require more robust approaches.

An auto scaling technique named CloudWatch [21] was enabled by Amazon to reduce the EDoS attack effects by monitoring the cloud resources. CloudWatch allows the customers to define the cloud platforms elasticity limits and thus reduce the EDoS attacks effects. However, CloudWatch will freeze the cloud service when elasticity reaches the predefined upper bound threshold. Thus, the legitimate users will not be able to access the cloud services until refreshing the quota which leads to similar behavior as DDoS attacks.

Self-verifying Proof of Work (sPoW) is an approach that focuses on proving client commitment through solving crypto-puzzles [22]. Initially, clients request cloud server access by choosing a crypto-puzzle with k bits difficulty level. Then, the server generates a crypto-puzzle to protect the connection server information such as IP address and port number. Finally, the puzzle requester running on the client-side brute forces the k bits to discover the server information. Upon succeeding in solving the puzzle, the server establishes a secure communication channel with the client. The sPoW has several shortcomings such as asymmetric power consumption problem due to generating and solving the puzzles by servers and clients, respectively, and puzzle's difficulty impact on false positives. Moreover, Gligor [23] stated that the client puzzles used as a proof of work are ineffectual as they force a high overhead on the legitimate users requests and offer extremely feeble assurances. Similar approaches to the sPoW technique are presented in [24], [25], [26] with similar limitations.

Al-Haidari et al. [27] proposed an EDoS mitigation technique called EDoS-Shield. The EDoS-Shield architecture consists of two main components. The first component is a virtual firewall with white and black lists which hold the IP addresses of clients targeting the cloud. The second component is a Verifier cloud Node (V-Node) that uses the Graphics Turing Test (GTT) [28]. The GTT generates visual tests to differentiate between requests sent by legitimate users and automated attackers. Subsequently, the V-Node updates the whitelist and the blacklist based on the verification process outcome. If a user passes the GTT, the user's IP address is held in the whitelist and subsequent requests from the same IP address are forwarded directly to the cloud. By contrast, if a user fails the GTT, the user's IP address is held in the blacklist and subsequent requests from this IP address are dropped by the firewall. One of the major shortcomings of this approach is its vulnerability

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2018.2805907, IEEE Transactions on Cloud Computing

to IP spoofing. This problem might cause an EDoS attack if an attacker spoofs the IP address to use an address already existing in the whitelist of the V-Node. Another disadvantage is the false positives, in which the EDoS-Shield may unknowingly block a NAT IP address that serves an entire network due to the misbehaving of one attacker that belongs to the same NAT-based network. Likewise, the problem of false negatives can occur when clients with IP addresses that belong to the whitelist become attackers of the cloud. Finally, the EDoS-Shield adds an overhead on the firewall when checking the incoming request IP address. This overhead affects the response time of the legitimate requests even under the normal behavior of the cloud.

An enhanced EDoS-Shield [29] attempts to solve the spoofed IP addresses problem in [27]. The technique appends a Time To Live (TTL) value to the received request IP address and adds a counter of unmatched TTL values to the white and black lists to decide if the request has a spoofed IP address. This technique has similar limitations to the EDoS-Shield [27]. In addition, cloud services are accessible from everywhere, so it is difficult to recognize clients TTL values. Moreover, attack tools that change the TTL value exist, so it is not always correct and cannot be trusted [30].

Masood et al. [31] proposed a mitigation technique called EDoS Armor against EDoS attacks targeting cloud ecommerce applications. The EDoS Armor has dual defense system; admission control and congestion control. The admission control is used to limit the number of clients who access the cloud services simultaneously, whereas the congestion control is used to assign priority to the permitted clients based on a browsing behavior learning mechanism. The EDoS Armor has many limitations such as restraining the cloud elasticity feature as the admission control limits the simultaneous user requests for cloud services. In addition, the average response time for legitimate clients is high because of the continuous learning and the priority updating. Another limitation is the false positives resulting from the possibility of blocking an entire NAT-based network.

Baig et al. [32] proposed a novel EDoS mitigation technique. The technique architecture consists of a virtual firewall, a load balancer, a database, and a Virtual Machine (VM) investigator. The technique performs extra analysis on all incoming requests when the cloud resources CPU utilization exceeds 80%. The technique uses a Turing test and a User Trust Factor (UTF) to determine the user's legitimacy. The technique randomly selects numbers, referred to as Random Check (RC), between 1 and a pre-defined allowable total requests per minute threshold. The count of the selected RC values is equal to a pre-defined allowable Concurrent Requests Per Second (CRPS) threshold. The request from a user who exceeds the CRPS threshold with low UTF is dropped. By contrast, the user's request is directly sent to the cloud when the user does not exceed the CRPS threshold, the user's request count does not match any of the RC values, and the UTF is good. Otherwise, the user is requested to solve a Turing test. The proposed technique has some limitations such as forcing legitimate users to respond to a Turing test even when their total request count is less than the CRPS. This occurs when the total request count matches one of the RC values. Further, the technique may unknowingly block an entire NAT-based network.

2.2 Methods to Distinguish Clients that belong to a NAT-based Network

Network Address Translation (NAT) devices are a convenient way to hide the source of malicious behaviors. In [33], the authors explore the use of a machine learning approach to detect the behavior of NAT devices using only network flows. The authors stated that using the TTL field to differentiate the types of NAT-based network users is not accurate. This is mainly because some NAT implementations choose not to decrement the TTL values, or that the TTL values can be modified by certain attack tools.

A system for defending against DDoS attacks is presented in [34]. The authors stated that the system helps in preventing a legitimate client from being blocked in a NAT-based network. The system uses a blacklist rule table to store the client IP address, the web server IP address, and the web server port when excessive web page request traffic is detected by a detection engine. The system that is located between the client and the web server intercepts incoming requests from the client. Subsequently, the system transmits to the client a Uniform Resource Locator (URL) redirection packet containing a virtual IP information of the corresponding web server. Accordingly, the system determines that a client who submits a request is a legitimate client when the client submits requests that use the received virtual IP information. Otherwise, the system considers the client malicious when it fails to use the virtual IP information, and adds the client IP address to the blacklist rule table. A major drawback of the proposed system is associated with the system's blacklist rule table not recording the incoming requests' source port numbers. As such, requests received from legitimate and malicious clients that belong to the same NAT-based network share the NAT public IP address. Hence, the system inadvertently blocks an entire NAT-based network. This occurs when the NAT IP address is added to the blacklist rule table as a result of a malicious client failing to use the virtual IP information sent by the system.

3 The Proposed Approach

The EDoS-ADS technique uses the threshold and the duration parameters that are typically associated with the cloud auto scaling conditions. The EDoS-ADS depends on the use of an average CPU utilization threshold as a parameter to trigger the auto scaling condition since it is a good indicator of abnormal behavior. A duration value is also used by EDoS-ADS to ensure that auto scaling is triggered only for legitimate requests in order to mitigate EDoS attack effects. So, the EDoS-ADS is considered a reactive technique because it starts running only when the threshold parameter is crossed at which time it verifies whether the incoming requests are from legitimate users or from attackers. Thus, it only allows the legitimate users requests to access the cloud and drops all attack traffic. Moreover, to overcome the effect of fluctuations in the cloud resources average CPU utilization, four thresholds with two timers are used. The thresholds are scaling-up upper, scaling-up lower, scaling-down upper, and scaling-down lower thresholds with values of 80%, 75%, 30%, and 35%, respectively [35]. The auto scaling timers are the upper threshold and the lower threshold timers with assumed durations of 5 minutes and 1 minute,

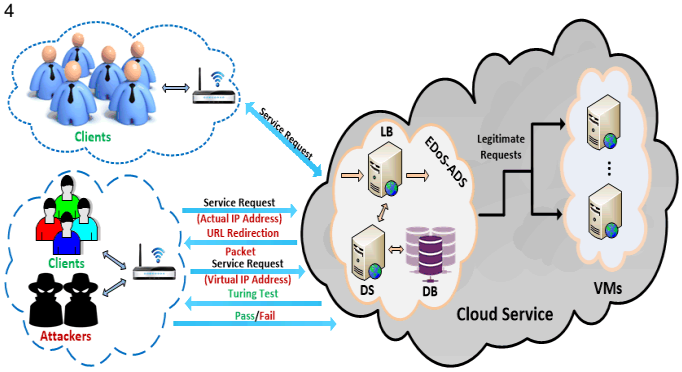

Fig. 1. Architecture of EDoS-ADS.

respectively [32]. The benefit of using these thresholds and timers is allowing auto scaling to be triggered but not be affected by momentary fluctuations in the utilization.

The EDoS-ADS has four cloud modes; ***normal***, ***suspicion***, ***flash overcrowd***, and ***attack***. The ***normal mode*** indicates that the cloud CPU utilization due to all incoming requests is below the scaling-up utilization thresholds. The mode switches to the ***suspicion mode*** when the cloud CPU utilization exceeds the scaling-up utilization thresholds. If the increase in the incoming requests is classified as coming from attackers, the mode switches to the ***attack mode***. If not, the mode switches to the ***flash overcrowd mode*** indicating that the incoming requests are from legitimate clients.

The EDoS-ADS main components are the **Load Balancer (LB)**, the **DataBase (DB)**, and the **Defense Shell (DS)** as shown in Fig. 1. The DS works differently in the ***suspicion mode*** than in the ***attack mode***. For simplicity, the terms **Suspicion Shell** and **Attack Shell** refer to the DS when working in the ***suspicion mode*** and the ***attack mode***, respectively. Details of the EDoS-ADS components are presented next.

Load Balancer (LB): The LB ensures an even distribution of the incoming requests among all active cloud servers [36]. The LB monitors the current utilization of a cloud system and the auto scaling utilization thresholds. Then, the LB redirects the incoming requests to the cloud servers or to the DS depending on the current mode and the current CPU utilization level. The cloud starts operating in the ***normal mode*** during which the LB forwards all incoming requests to the cloud servers. The cloud responds to the clients' requests without any overhead or checking. The cloud remains in the ***normal mode*** as long as the average CPU utilization does not exceed the scaling-up upper threshold. However, once the average CPU utilization surpasses the scaling-up upper threshold, referred to as the suspicion region, then the cloud mode switches to ***suspicion mode*** and all incoming requests are directed to the **Suspicion Shell** for further investigation.

DataBase (DB): The DB has one table referred to as the Behavior table. The Behavior table is used by the DS to track the clients past behavior. In addition, it is used to assist the DS in determining whether the cloud is under an ***attack mode*** or a ***flash overcrowd mode***. As such, EDoS-ADS stores the client status, Concurrent Requests Per Second (CRPS), Last Seen, IP address, source port number, and Trust Factor (TF) in the DB. The DS uses the IP address and port number as the client key. The status field reflects the client current status in the cloud which can be one of three types; *Legitimate*, *Suspicious*, or *Malicious*. The CRPS counts the total number of client requests received by the cloud in one second. The Last Seen holds the time of the last request sent by the client and it is used by the DS to update the CRPS value as explained later in the DS description.

The DB keeps track of the source IP address and port number of each request sent by a client. As a result, the DB assists the EDoS-ADS in distinguishing between legitimate users and attackers especially when both belong to a NAT-based network. Such a distinction helps prevent the blocking of an entire NAT-based network. Note that a NAT replaces the source IP address and port number of each outgoing packet with the NAT public IP address and a randomly chosen unused port number, respectively. As such, the port number assigned by the NAT can be used to distinguish between different clients that belong to a NAT-based network since it is impossible to assign the same port to several active users. Thus, the EDoS-ADS uses the source IP address and port number that are stored in the DB to address the problems of IP spoofing, blocking of NAT-based networks, and distinguishing the different types of users.

The TF that is stored in the DB is a value between 0 and 1 assigned to each client depending on the client's response to the Graphics Turing Test (GTT) [28]. The TF value is set by the **Suspicion Shell** when a request from a new client passes the DS. A default TF value of 0.5 is used. The TF value is incremented or decremented with each client's response to subsequent GTTs. Note that the amount of TF increment should be less than the amount of TF decrement as the failure to respond to the GTT is a possible indicator of an attack [37]. Both the client status and TF are affected by the results of the GTT. If a new client fails the GTT, the TF value is decremented and the client's status is set to *Suspicious*. By contrast, if the new client passes the GTT, the TF value is incremented and the client's status is set to *Legitimate*. In [38], the authors classified trust factors into three different levels each with a specific interval as follows: *low* TF is [0, 0.25), *average* TF is [0.25, 0.75], and *good* TF is (0.75, 1]. The EDoS-ADS uses these levels to check the client's legitimacy. Clients with *good* TF level are considered trust worthy.

Defense Shell (DS): The DS is responsible for differentiating between legitimate users and attackers using two techniques; GTT [28] [39], and URL redirection [40]. The DS drops a request in case of a failure to respond to either a GTT or a URL redirection, and considers the request generator as an attacker. Note that by continuously sending GTT to legitimate clients they will likely turn away from using the cloud services. Hence, the DS uses the URL redirection scheme to resolve such a problem as explained next.

The URL redirection occurs when a client sends an HTTP request to access a web resource located at a specific URL. Then, the requested web server sends an HTTP response to the client browser with a location field in the header. This implies that the client must redirect to a different URL by using a virtual IP address [41]. Hence, the DS uses URL redirection to distinguish between legitimate clients and attackers. In particular, the DS considers the client as legitimate if the client request includes the cloud server virtual IP address that is used by the URL redirection technique. By contrast, an attacker typically runs a script on compromised machines to generate a large number of requests that use the cloud server actual IP address. Subsequently, the attacker usually does not wait for the cloud response packet. Thus,

the attacker fails to perform URL redirection and continues to use the server actual IP address in subsequent requests.

Upon receiving a request, the DS updates the values of Last Seen and CRPS associated with the client in the Behavior table. The difference between the client current time and Last Seen time is checked when a new request from a client arrives at the DS. If the time difference is less than or equal to 1 second, the CRPS value is incremented by 1. Otherwise, if the time difference is more than 1 second, the CRPS value is reset to 1. Then, Last Seen is updated to the current time. In addition, the DS uses the Total Request Count (TRC) parameter to count the total number of received requests. The TRC value is incremented upon receiving a request, and it is used later to determine the next cloud mode. The DS works differently in the **Suspicion Shell** than in the **Attack Shell** as explained next.

3.1 Suspicion Shell

The EDoS-ADS uses the **Suspicion Shell** when it suspects that the cloud is under attack after observing that the CPU utilization exceeds the scaling-up utilization threshold. As such, the **Suspicion Shell** checks if the client CRPS exceeds the Maximum Requests Per Second (MRPS) threshold set by the cloud adopter. This results in the following two cases:

Case 1 (CRPS $\leq$ MRPS): In such a case, the **Suspicion Shell** sends a URL redirect packet to the client if the client is sending a new request, and the request uses the cloud server actual IP address. Later, if the client is legitimate then the client uses the virtual IP address contained in the URL redirect packet for the current request and for all future requests. Thus, requests of such client are immediately forwarded to the cloud without the need to send a GTT since the requests use the server virtual IP address. Hence, the response time experienced by such a client is smaller than the case when the client must solve a GTT. Note that the URL redirection helps defend against attacks initiated by smart attackers who can guess the MRPS value and send requests without exceeding that value. Thus, the **Suspicion Shell** protects the cloud by dropping such attack requests.

Case 2 (CRPS $>$ MRPS): In such a case, the **Suspicion Shell** takes the same actions taken in **Case 1** if the client has a *good* Trust Factor (TF). On the other hand, the **Suspicion Shell** sends a GTT to a client sending a new request to the cloud if the client's TF is either *average* or *low*. The GTT helps distinguish between a legitimate client and an attacker. Specifically, a legitimate client will successfully reply to the GTT, and the client's request is then forwarded to the cloud. With every successful GTT reply, the legitimate client's TF improves until it becomes *good* at which point the client's subsequent requests are handled following the same actions taken in **Case 1**. By contrast, an attacker will fail to reply to the GTT. As such, the attacker requests are blocked from reaching the cloud, and the attacker's TF is decreased.

The **Suspicion Shell** counts the number of failed responses and records the total in the Malicious Request Count (MRC) parameter. The MRC is incremented by one for each failed GTT, and for each failed URL redirection. The MRC and TRC values are used to determine the next cloud mode as explained next. The **Suspicion Shell** is triggered for a period equal to the upper threshold timer. This period is referred to as the *Suspicion Timer*. If the *Suspicion Timer* does not expire, and the current system utilization decreases to less than the scaling-up lower threshold or it is in between the scaling-up upper and lower thresholds for more than the lower threshold timer, the cloud mode switches to ***normal***. By contrast, when the *Suspicion Timer* expires, the **Suspicion Shell** changes the mode to either a ***flash overcrowd*** or an ***attack*** depending on the percentage of malicious responses that is calculated by dividing the MRC over the TRC. If this value is below or equal to 8%, the cloud mode is set to ***flash overcrowd***. Otherwise, the mode is set to ***attack***. Note that the value of 8% is selected as a threshold for mode determination based on the work reported in [39]. If the cloud mode is set to ***flash overcrowd***, new VM instances are added, and a *Provisioning Timer* starts. The cloud remains in that mode for the timer duration. With the newly allocated VM instances, the cloud eventually returns to the ***normal mode*** [42]. During the ***flash overcrowd mode***, all requests are directly served by the cloud as in the ***normal mode***.

For early ***attack mode*** detection, the **Suspicion Shell** checks the past behavior for clients who failed the GTT. A client with *Malicious* status, *low* TF, and a CRPS more than MRPS is marked as an attacker and the cloud mode changes to ***attack***. In addition, the **Suspicion Shell** checks the TF level and the URL redirection ability for clients targeting the cloud every 1 minute [32]. This period is referred to as the *Flash Timer*. If clients with CRPS less than or equal to MRPS were able to redirect, and clients with CRPS larger than MRPS who have a *good* TF level and they redirected successfully, the cloud mode immediately becomes ***flash overcrowd*** without waiting until the *Suspicion Timer* expires. By contrast, if the previous condition is not met, the **Suspicion Shell** restarts the *Flash Timer*.

3.2 Attack Shell

The EDoS-ADS ultimately switches the cloud mode to either ***flash overcrowd*** or ***attack*** depending on the calculated percentage of malicious responses as stated in subsection 3.1. If the cloud mode switches to ***attack***, then the **Attack Shell** is activated. Accordingly, each client is asked to use the cloud server virtual IP address when sending requests. Specifically, the **Attack Shell** sends a URL redirect packet to each client that uses the cloud server actual IP address. Hence, two cases are considered:

Case 1 (Client uses virtual IP address): In such a case, the **Attack Shell** correctly identifies the client as legitimate. Moreover, the client request is forwarded to the cloud if the client CRPS does not exceed a dynamic threshold referred to as the *Allowable-RPS*. The *Allowable-RPS* is computed using the client's current TF value as follows [43]

$$\textit{Allowable-RPS} = \left\lfloor 0.444516 \times (81)^{TF} \right\rfloor \quad (1)$$

By contrast, the **Attack Shell** sends GTT to the legitimate client if the client CRPS exceeds the client's *Allowable-RPS* threshold. In such a case, the GTT helps verify the client legitimacy, and aids in improving the legitimate clients TF.

Case 2 (Client fails to use virtual IP address): In such a case, the client fails to perform URL redirection. Hence, the **Attack Shell** considers the client as an attacker, and drops the client request. Note that, as explained in the DS

description in section 3, legitimate clients can successfully perform URL redirection and use the cloud server virtual IP address. By contrast, attackers use scripts that typically do not wait for cloud response packets. Hence, attackers will fail to use the cloud server virtual IP address.

The cloud remains in the ***attack mode*** for a certain amount of time that is referred to as the *Attack Timer*. To select the *Attack Timer* period, the average duration of attacks found in the literature is considered. Both [44] and [45] found out that DoS and DDoS attacks lasted for less than 30 minutes. So, the Attack Period Timer (APT) of 30 minutes is chosen. Then, the *Attack Timer* is calculated by multiplying the APT value by the percentage of malicious responses measured during the ***suspicion mode***. Subsequently, if the *Attack Timer* expires and the percentage of malicious responses is greater than 8%, the cloud stays in the ***attack mode***. Further, the **Attack Shell** restarts the *Attack Timer*, and resets the TRC and MRC. By contrast, if the percentage of malicious responses is below or equal to 8%, the current mode is set to either ***suspicion mode*** or ***normal mode*** depending on the current CPU utilization. If the CPU utilization is within the suspicion region, the cloud mode switches to ***suspicion mode***, else it returns to ***normal mode***.

Note that the difference between the **Suspicion Shell** and the **Attack Shell** is that, unlike the **Attack Shell**, the **Suspicion Shell** does not limit the number of requests a legitimate client can submit per second. This is because EDoS-ADS uses the **Suspicion Shell** when it is unsure if the cloud is currently receiving *flash overcrowd* or *attack* traffic. Once EDoS-ADS decides that the cloud is under attack it uses the **Attack Shell**, and for a legitimate client with TF equal to 1 it cautiously limits the number of requests the client can submit to 36 requests per second according to (1).

3.3 Proposed Approach Validation

The cloud can be in one of four modes: ***normal***, ***suspicion***, ***flash overcrowd***, or ***attack***. If the cloud mode is ***normal*** or ***flash overcrowd***, then all incoming requests are immediately forwarded to the cloud. Otherwise, if the cloud mode is ***suspicion*** or ***attack***, then the **Suspicion Shell** and the **Attack Shell** require each client to either use the cloud server virtual IP address or solve a GTT before forwarding the client request to the cloud. As stated in subsections 3.1 and 3.2, legitimate clients, unlike attackers, are able to use the cloud server virtual IP address and to solve GTTs. Thus, all legitimate clients requests will be forwarded to the cloud while all attack requests will be blocked. Note that since EDoS-ADS uses both the IP address and port number to identify a client, EDoS-ADS effectively handles IP spoofing, and ensures that legitimate clients requests are not blocked.

4 Simulator

In this section, we discuss the simulator used to evaluate the EDoS-ADS. We present also the measured performance metrics in the simulation, and describe the EDoS-ADS simulation model and its verification and validation.

4.1 Simulator Design and Modeling

The cloud environment is simulated using CloudSim [46]. CloudSim is extended through this work to support auto scaling, and can be accessed at [47]. For the clients requests, we utilize the Poisson traffic model for the EDoS attack [48]. In addition, we consider a single-class service where all cloud clients' requests have the same processing procedure as discussed in [27]. A cloud usually has multi cloud servers offering the service to the cloud customers. Thus, a parallel *M/M/1* queuing model is used for the cloud service [49]. We use *M/M/1* queuing model with the VM and the load balancer instances as the cloud has large finite buffers and thus the probability of a buffer overflow is negligible [50].

4.2 Performance Metrics

Simulation experiments are conducted to simulate a cloud under an EDoS attack to evaluate the cloud performance with and without the use of a mitigation technique. The key performance metrics considered are the cloud response time, computing resources utilization, and throughput. Further, the cost related to the cloud computing resources and bandwidth allocations is measured.

The average response time is measured by averaging the residence time, which is the departure time ($D\tau$) minus the arrival time ($A\tau$), for all N served requests as

$$RT_{Avg} = \frac{1}{N}\sum_{i=1}^{N}\left(D\tau_i - A\tau_i\right) \tag{2}$$

We assume that all cloud servers have the same capacity of computing power, $\mu_i = \mu$, and the requests' arrival rate at each server is $\lambda_i = \lambda/S$, where λ is the total traffic arrival rate, and S is the number of running VM instances. Hence, the mean computing utilization is

$$\rho = \lambda/(S \times \mu) \tag{3}$$

where ρ is the cloud server utilization, and μ is the cloud server service rate. The CPU utilization of each server is calculated by dividing the total server processing time by the total simulation time. Hence, the average CPU utilization is calculated as

$$\rho_{Avg} = \frac{1}{S}\sum_{i=1}^{S}\frac{Processing\ Time_i}{Simulation\ Time_i} \tag{4}$$

The cloud throughput can be calculated directly from Little's formula as in (5), while the throughput of each cloud server is measured by dividing the number of served requests, N, by the simulation time. The cloud service throughput is computed by taking the aggregated throughput of all servers in the auto scaling group according to (6).

$$Throughput = \mu \times \rho = \lambda/S \tag{5}$$

$$Measured\ Throughput = \sum_{i=1}^{S}\frac{N_i}{Simulation\ Time_i} \tag{6}$$

Amazon Web Services (AWS) pricing model [51] is used to compute the cost. The cost is given by Al-Haidari et al. [12] as

$$Cost = \left(Price_{BW} \times \lambda + Price_{VM} \times \rho \times S\right) \times T \tag{7}$$

where $Price_{BW}$ is the price per input/output GB, λ is the arrival rate in GB/hr., $Price_{VM}$ is the price of using the computing resources per hour, ρ is the average CPU utilization

of all computing resources during the total running time T, and S is the number of running VM instances.

4.3 Simulation Setup

Several experiments are conducted while using the same assumptions used by the EDoS-Shield work [27]. Table 1 summarizes the parameters used in the simulation.

4.4 Simulator Validation

The simulation is validated by comparing obtained EDoS-Shield CloudSim simulation results with the EDoS-Shield work results [27]. The results show that both simulations are similar for the response time with an error less than 0.48%. For the CPU utilization, the results for both the EDoS-Shield CloudSim simulation and the EDoS-Shield work are identical with an error less than 0.075%. The detailed validation results are intentionally omitted for brevity and can be found at [52]. Further, the EDoS-Shield simulation results were validated by conducting an experimental testbed [53].

TABLE 1
Simulation Parameters

Parameter	Value	Reference
Auto scaling metric	CPU usage	[27]
Scaling-up upper threshold	80%	[27], [35]
Scaling-up lower threshold	75%	[35]
Scaling-down upper threshold	30%	[35]
Scaling-down lower threshold	35%	[32], [35]
Upper threshold duration	5 min	[27], [32]
Lower threshold duration	1 min	[32]
Cloud instance service rate	100 Req/sec	[54]
Cloud instance cost	\$0.115/hr	[27]
Initial running servers	5	[27]
Auto scaling size	2	[27]
Provisioning overhead	55.4 sec	[42]
Server response packet size	580 byte	[55]
Load balancer service time	5.8 μs	[56]
Flash Timer	1 min	[32]
APT	30 min	[44]
MRPS	4	[32]
Large instance cost	\$0.48/hr	[27]
Bandwidth allocation cost	\$0.01/GB	[51]
GTT response time	13.06 sec	[39]
URL redirection overhead	0.63 sec	[57]

5 Simulation Results and Analysis

In this section, the EDoS-ADS simulation results are presented. The number of requests that target the cloud is selected in such a way to ensure that the simulation reaches the steady state. In addition, each experiment is repeated 10 times and the results are averaged. Four experimental scenarios are conducted using the simulation model discussed in section 4. These scenarios study the ***normal mode*** (1 scenario), the ***flash overcrowd mode*** (1 scenario), and the ***attack mode*** (2 scenarios). The results are compared with the EDoS-Shield work [27]. For purposes of briefness, and given that the ***normal mode*** results show no overhead associated with the use of EDoS-ADS [52], only the ***flash overcrowd mode*** and the ***attack mode*** results are provided.

5.1 Flash Overcrowd Mode Results

The ***flash overcrowd mode*** occurs when there is a large amount of legitimate traffic referred to as flash traffic arriving at the cloud. Specifically, it happens when the current utilization exceeds the scaling-up utilization thresholds, and the percentage of malicious responses is below or equal to 8% as stated in section 3. Thus, in this scenario, the legitimate arrival rates are varied from 400 to 6400 requests per second. The simulation scenario objective is to check if the EDoS-ADS impacts the cloud performance during flash overcrowd. Further, this scenario shows the capability of the EDoS-ADS to auto scale when having flash traffic. Hence, the EDoS-ADS results are compared with the results of the EDoS-Shield, and the cloud that uses the auto scaling technique but does not use an EDoS mitigation technique. The latter scenario will be referred to as Auto Scaling Only (ASO). Note that different number of VM instances are allocated based on the arrival rates. The EDoS-Shield work analytically determines the VM instances as

$$VM_{required} = \left\lceil 1.25 \times \lambda/\mu + 1 \right\rceil \quad (8)$$

where λ is the total rate of incoming traffic, and μ is the service rate of the cloud server.

Accordingly, Fig. 2 compares the number of allocated VM instances for the considered cases. Note that the results reported by the EDoS-Shield work assume an ideal case where the newly allocated VM instances service the queued requests instantaneously. Therefore, the EDoS-Shield results can be considered optimal at best.

Fig. 3 shows the time needed to allocate the required number of VM instances for the CPU utilization to drop below 80%. Because of the EDoS-ADS fast detection of the ***flash overcrowd mode***, the EDoS-ADS reaches the desired number of VM instances with less time than the other cases.

Fig. 4 shows the CPU utilization evaluation for the different cases. The results are identical for EDoS-ADS and ASO as both techniques use the same number of VM instances to serve the legitimate clients requests. However, the EDoS-Shield CPU utilization is greater than the EDoS-ADS CPU utilization since EDoS-Shield uses less number of VM instances than EDoS-ADS while serving the same number of incoming requests.

The average response time evaluation is shown in Fig. 5. Overall, only a limited response time increase is observed with the increase of the arrival traffic. The limited response time increase is due to the auto scaling mechanism that allocates sufficient VM instances to process the additional load. Further, the response time results are almost identical

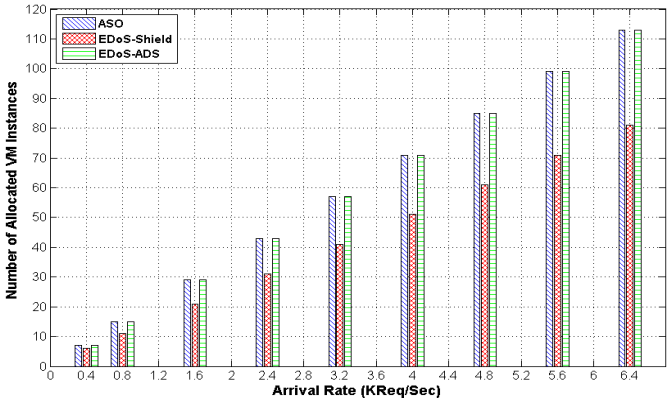

Fig. 2. Number of allocated VM instances (flash overcrowd mode).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2018.2805907, IEEE Transactions on Cloud Computing

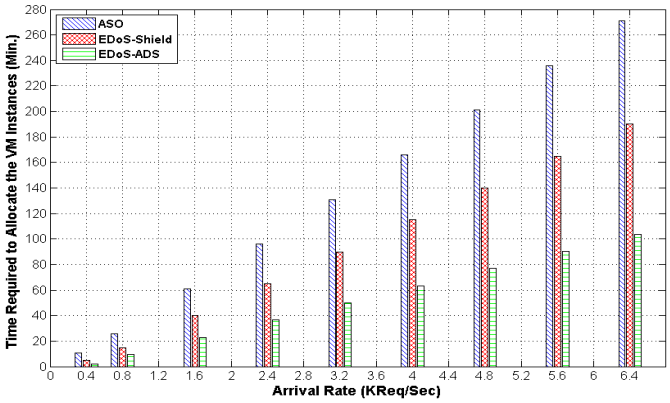

Fig. 3. Time to allocate the VM instances (flash overcrowd mode).

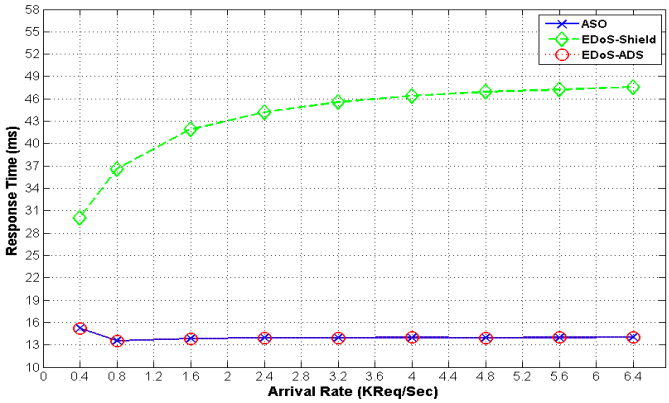

Fig. 5. Response time (flash overcrowd mode).

for EDoS-ADS and ASO as both use the same number of VM instances for the different arrival rates. By contrast, the EDoS-Shield response time is greater than the EDoS-ADS response time due to using fewer number of VM instances which results in large queuing delays which in turn causes higher response time. Further, the EDoS-Shield has a higher response time due to the extra overhead on the EDoS-Shield firewall as it must check the IP address of every request.

The throughput evaluation is shown in Fig. 6. The throughput results are identical for all considered cases. This is due to having sufficient on-demand VM instances to handle all traffic including the flash traffic. In addition, the results show that there is no impact on the throughput.

Fig. 7 compares the cost for the different cases. We considered the cost of the VM instances and the bandwidth allocation for 10 hours [5]. The costs are identical for EDoS-ADS and ASO as both cases allocate the same number of VM instances, and both cases have the same CPU utilization. Although EDoS-Shield allocates less number of VM instances than EDoS-ADS, the EDoS-Shield cost is almost identical to the EDoS-ADS cost. This is because the EDoS-Shield CPU utilization is greater than that of EDoS-ADS.

5.2 Attack Mode Results

In this mode, both legitimate clients and attackers are considered. The purpose of this setup is to evaluate the EDoS-ADS effectiveness when the cloud is under an attack. Two scenarios are considered during the ***attack mode***. Scenario one assumes that legitimate clients and attackers are hosted in non NAT-based networks. By contrast, scenario two considers having legitimate clients and attackers hosted by the same NAT-based network. Recall from subsection 3.2 that during an attack, the **Attack Shell** forces each client to use the cloud server virtual IP address before the client request is forwarded to the cloud. If the client successfully uses the virtual IP address, the client is considered to be legitimate and the request is forwarded to the cloud. If not, the client is considered to be an attacker and the request is dropped.

5.2.1 *Users do not belong to NAT-based Networks*

In this scenario, the attack rates are varied from 400 to 6000 requests per second. Further, the legitimate arrival rate is assumed to be fixed at 400 requests per second, and each legitimate client transmits at a rate less than the *Allowable-RPS*. The EDoS-ADS results are compared with the results of the EDoS-Shield and the Auto Scaling Only (ASO). For every attack rate, the initial number of VM instances used is set to 7, 6, and 7 for the EDoS-ADS, EDoS-Shield, and ASO, respectively, following the results of Fig. 2. Further, to assess the EDoS-ADS performance overhead, the EDoS-ADS performance results are reported for the case when the cloud is under attack and the case when it is not under attack. The latter case is referred to as EDoS-ADS (No EDoS Attack).

Three cases for the EDoS-Shield are considered; EDoS-Shield (optimal case), EDoS-Shield (whitelist case), and EDoS-Shield (blacklist case). The EDoS-Shield (optimal case) refers to the case when there is no spoofing of IP addresses. Thus, the legitimate clients IP addresses are in the whitelist and the attackers IP addresses are in the blacklist. By contrast, the EDoS-Shield (whitelist case) refers to the case when the attackers use spoofed IP addresses that are already in the whitelist. Hence, the EDoS-Shield considers

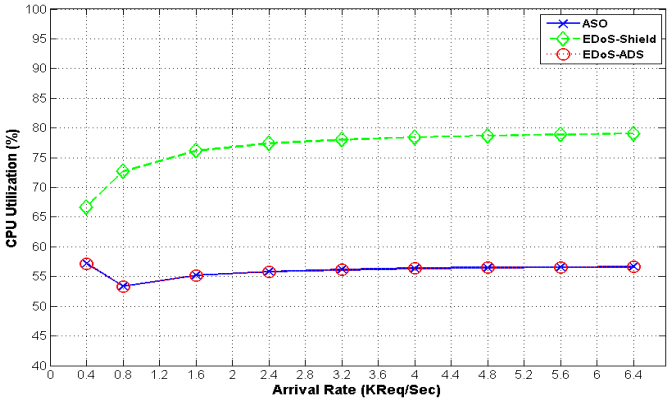

Fig. 4. CPU utilization (flash overcrowd mode).

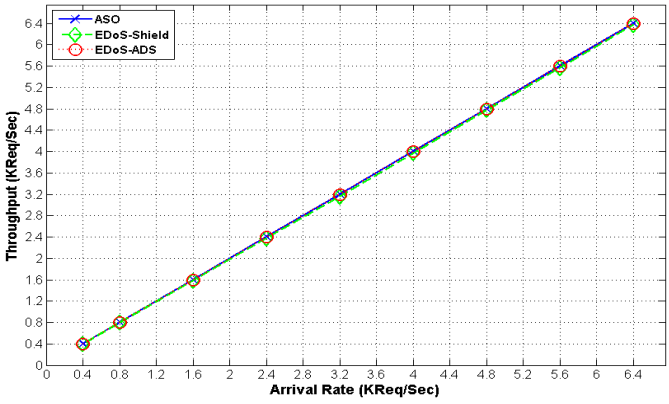

Fig. 6. Throughput (flash overcrowd mode).

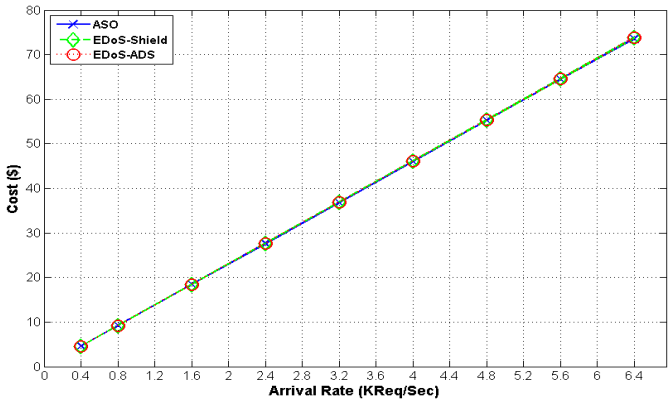

Fig. 7. Cost (flash overcrowd mode).

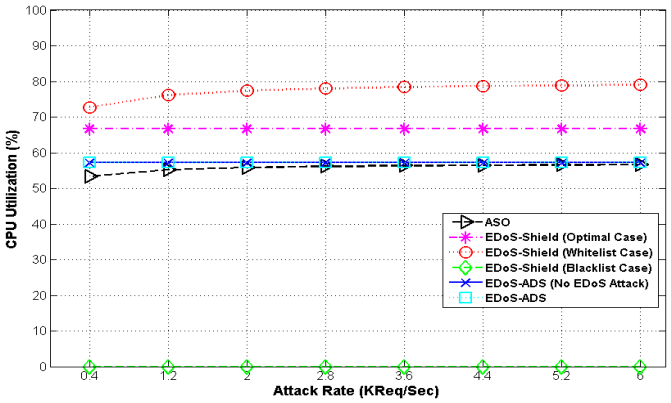

Fig. 9. CPU utilization (different NAT-based networks).

these attackers as legitimate clients, and the corresponding traffic is forwarded to the cloud. Finally, the EDoS-Shield (blacklist case) is used to describe the EDoS-Shield when the attackers carry out an attack using spoofed IP addresses that are currently neither in the whitelist nor in the blacklist. As such, the spoofed IP addresses end up in the blacklist. Thus, the EDoS-Shield considers the legitimate clients who are the actual owners of the spoofed IP addresses as attackers, and such clients are, subsequently, not able to access the cloud because their IP addresses are already in the blacklist.

Fig. 8 shows a comparison of the number of allocated VM instances. Both EDoS-Shield (whitelist case) and ASO have allocated higher number of VM instances than the other cases as these two cases consider the attack traffic as legitimate traffic. As such, the two cases auto scale to serve all incoming traffic. By contrast, EDoS-Shield (optimal case) uses the initial number of VM instances to serve only the legitimate traffic. Similarly, EDoS-Shield (blacklist case) uses only the initial number of VM instances for all attack rates considered, and does not perform auto scaling. This is because the legitimate clients and attackers were unable to access the cloud since their IP addresses are already in the blacklist. Finally, with EDoS-ADS, the cloud mode changes to ***suspicion mode*** when the CPU utilization exceeds the scaling-up upper threshold as stated in section 3. As such, the EDoS-ADS Load Balancer (LB) component redirects all incoming requests to the **Suspicion Shell**. Accordingly, the **Suspicion Shell** sends a Graphics Turing Test (GTT) and URL redirection packets to the clients to decide if the cloud is in ***flash overcrowd mode*** or ***attack mode***. Due to the **Suspicion Shell** fast detection of the EDoS attack, the mode changes to ***attack mode***. Hence, the **Attack Shell** is activated which enforces the use of the cloud server virtual IP address for all incoming requests. Recall from subsection 3.2 that legitimate clients successfully use the server virtual IP address, while attackers fail to do so. Thus, EDoS-ADS blocks all attack traffic while forwarding the legitimate clients traffic to the cloud. Hence, EDoS-ADS uses only the initial number of VM instances whether there is an attack on the cloud or not.

The CPU utilization is shown in Fig. 9. As discussed in subsection 3.2, the EDoS-ADS activates the **Attack Shell** during an attack to block all attack requests. Thus, the EDoS-ADS results are constant and identical whether there is an attack on the cloud or not as in both cases EDoS-ADS uses the VM instances to serve only the legitimate clients requests. Conversely, EDoS-Shield (optimal case) CPU utilization is greater than that of EDoS-ADS since it uses less number of VM instances than EDoS-ADS. By contrast, EDoS-Shield (whitelist case) considers the attack traffic as legitimate traffic. Hence, EDoS-Shield (whitelist case) CPU utilization is the highest and increases as the attack rate increases. Finally, as ASO allocates the highest number of VM instances as shown in Fig. 8, it results in CPU utilization that is initially lower than that of EDoS-ADS. As the attack rate increases, ASO serves more attack traffic while EDoS-ADS blocks such traffic. Thus, ASO CPU utilization becomes almost equal to that of EDoS-ADS. Note that EDoS-Shield (blacklist case) CPU utilization is zero as the legitimate clients requests are dropped since their IP addresses are already blacklisted.

Fig. 10 presents the average response time evaluation.

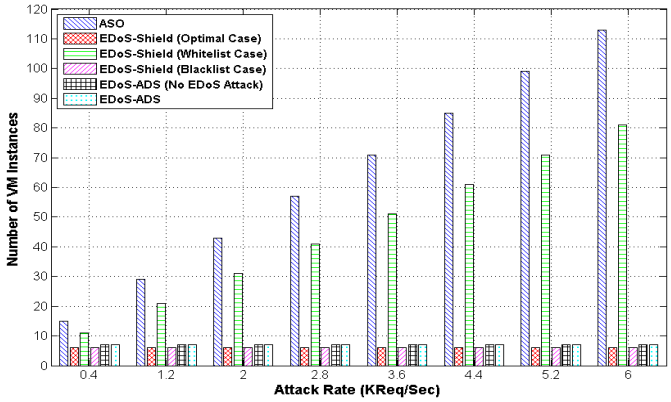

Fig. 8. Number of allocated VMs (different NAT-based networks).

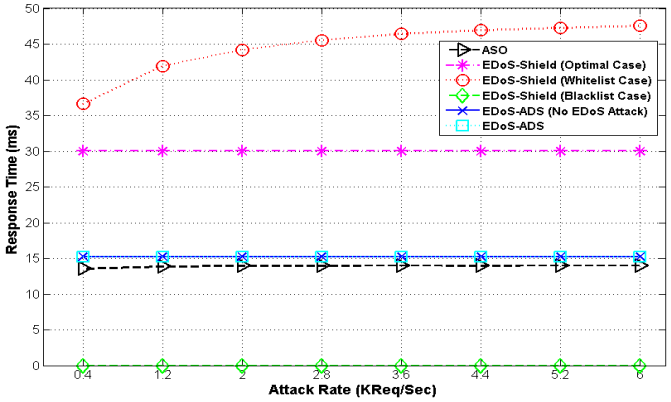

Fig. 10. Response time (different NAT-based networks).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2018.2805907, IEEE Transactions on Cloud Computing

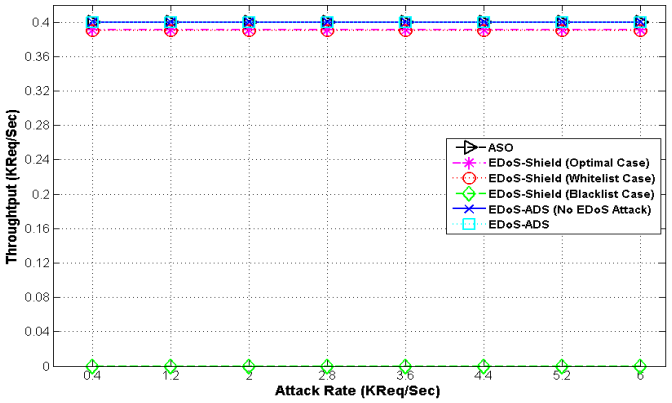

Fig. 11. Legitimate requests throughput (different NAT-based networks).

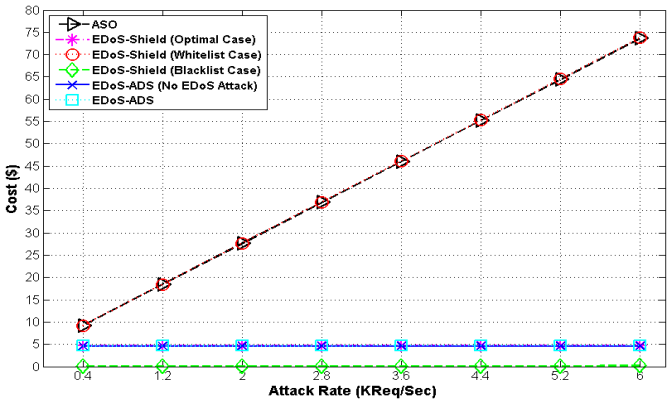

Fig. 12. Cost (different NAT-based networks).

The EDoS-ADS response time results are constant and identical whether there is an attack traffic or not. Similarlry, the EDoS-Shield (optimal case) response time results are constant for all attack rates. Hence, the legitimate users response time is unaffected by the attack traffic. However, EDoS-Shield (optimal case) response time is greater than that of EDoS-ADS due to the different number of allocated VM instances. As stated in subsection 5.1, the fewer number of VM instances in EDoS-Shield (optimal case) results in large queuing delays which in turn causes higher response times. For EDoS-Shield (whitelist case) and ASO, the response time slightly increases with increasing attack traffic. This is due to the delay associated with the auto scaling mechanism that allocates more VM instances to serve both legitimate and attack traffic. Based on Fig. 8, ASO allocates the highest number of VM instances and that results in lowering the response time to be slightly lower than that of EDoS-ADS. Finally, EDoS-Shield (blacklist case) response time is zero as all requests are dropped since the legitimate clients IP addresses are already blacklisted.

The legitimate requests throughput is shown in Fig. 11. As stated in subsection 3.2, the **Attack Shell** is used by EDoS-ADS to block all attack requests. Thus, the EDoS-ADS results are constant and identical whether the cloud is under attack or not. Likewise, the ASO results are identical to the EDoS-ADS results as both cases have sufficient VM instances allocated to serve the incoming requests. Moreover, the EDoS-ADS and ASO results show that the legitimate requests throughput is equal to the legitimate arrival rate of 400 requests per second and is unaffected by the attack. By contrast, the results show slight degradation in EDoS-Shield (optimal and whitelist cases) throughput as reported in [27]. Finally, the EDoS-Shield (blacklist case) throughput is zero as all requests are considered to be attack requests and are blocked by the EDoS-Shield firewall.

Fig. 12 compares the EDoS-ADS cost with the other cases assuming that the attack lasts for 10 hours [5]. As explained in subsection 3.2, EDoS-ADS uses the **Attack Shell** to block all attack requests. Thus, the EDoS-ADS costs are constant and identical whether there is an attack traffic or not. This is because both cases allocate the same number of VM instances, and have the same average CPU utilization. Further, EDoS-Shield (optimal case) cost is identical to EDoS-ADS cost even though it allocates less VM instances than EDoS-ADS. This is because EDoS-Shield (optimal case) CPU utilization is greater than that of EDoS-ADS and, according to (7), the cost will be the same. Finally, the EDoS-Shield (whitelist case) and ASO costs are greater than EDoS-ADS cost as they allocate higher number of VM instances than EDoS-ADS. Note that EDoS-Shield (blacklist case) cost is zero since the CPU utilization is zero too.

To summarize the results of this simulation scenario, EDoS-ADS effectively blocks all attack requests while forwarding all legitimate requests to the cloud. In doing so, EDoS-ADS uses the least amount of VM instances needed to handle only legitimate requests. Moreover, EDoS-ADS outperforms both EDoS-Shield and ASO with respect to CPU utilization, legitimate requests response time and throughput, and cost while making sure that such performance metrics are unaffected by the attack requests.

5.2.2 *Users belong to Same NAT-based Network*

This scenario assumes that legitimate clients and attackers belong to the same NAT-based network. Thus, the legitimate clients and attackers IP addresses appear to the cloud to be the same and equal to the NAT router public IP address. This scenario is extremely important as it reflects a real-life scenario, and it is more serious than the previous scenario since it can result in blocking an entire NAT-based network. Note that in this scenario EDoS-ADS is able to distinguish between different clients since it uses the client IP address and port number as the client key as stated in section 3.

The EDoS-Shield is vulnerable to blocking an entire NAT-based network due to an EDoS attack by attackers that belong to a NAT-based network. Thus, two cases for the EDoS-Shield are considered; the (whitelist case), and the (blacklist case). The EDoS-Shield (whitelist case) assumes that the NAT public IP address is already in the whitelist. Therefore, the EDoS-Shield considers the attackers that belong to the NAT-based network as legitimate clients and it allows them access to the cloud. By contrast, the EDoS-Shield (blacklist case) assumes that the NAT public IP address was not used earlier to access the cloud. Thus, when an attacker that belongs to the NAT-based network launches an EDoS attack, the EDoS-Shield adds the NAT public IP address to its blacklist. As such, legitimate clients that belong to the NAT-based network are, subsequently, unable to access the cloud because, from the cloud point of view, their IP addresses are already blacklisted.

In this scenario, a fixed legitimate traffic load that consumes 40% of the CPU utilization is generated [58]. The

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2018.2805907, IEEE Transactions on Cloud Computing

TABLE 2
Clients Targeting the Cloud

Set ID	Type	Number of Requests Sent	Number of Clients
1	Legitimate	CRPS = 1 < MRPS	100
2	Legitimate	CRPS = 5 > MRPS	20
3	Malicious	CRPS = 1 < MRPS	100
4	Malicious	CRPS = 4 = MRPS	25
5	Malicious	CRPS = 10 > MRPS	10

arrival rate to the cloud is divided into five group sets as shown in Table 2 with a total legitimate arrival rate of 200 requests per second. The simulation uses 5 VM instances as an initial cloud setup. The attack traffic is assumed to start after one minute of simulation. The EDoS-ADS results are compared to the ASO results. Note that the EDoS-Shield (whitelist case) has similar results to ASO results while the EDoS-Shield (blacklist case) blocks the entire NAT-based network as explained in subsection 5.2.1. Accordingly, the EDoS-Shield results are not presented.

Fig. 13 shows a comparison of the number of allocated VM instances for EDoS-ADS and ASO. It is clear that ASO has two additional VM instances allocated as compared to EDoS-ADS because ASO considers the attack traffic as legitimate. As a result, ASO auto scales to serve all incoming traffic. By contrast, EDoS-ADS uses the **Attack Shell** to properly identify the attack traffic and blocks it from reaching the cloud. Thus, the EDoS-ADS uses just the initially allocated VM instances to serve only the legitimate requests.

The CPU utilization is presented in Fig. 14. Both EDoS-ADS and ASO have an average CPU utilization of 40% during the first minute of simulation. Note that during the first minute, only legitimate traffic is received by the cloud, and the 40% CPU utilization is a result of serving that traffic. As soon as the attack starts, the EDoS-ADS and ASO CPU utilization rises to 100%. Starting from minute six, ASO CPU utilization returns to about 72% after the cloud auto scales.

Recall from subsection 5.2.1 that once EDoS-ADS changes the cloud mode to ***attack mode*** it activates the **Attack Shell** which enforces the use of the cloud server virtual IP address for all incoming requests. Thus, between minutes one and two in Fig. 14, the **Attack Shell** asks the second group of legitimate clients to solve a GTT although the incoming requests use the server virtual IP address. This is because the second group clients request rate is higher than their *Allowable-RPS*. Such clients require 13.06 seconds on average to solve the GTT. Thus, the second group traffic

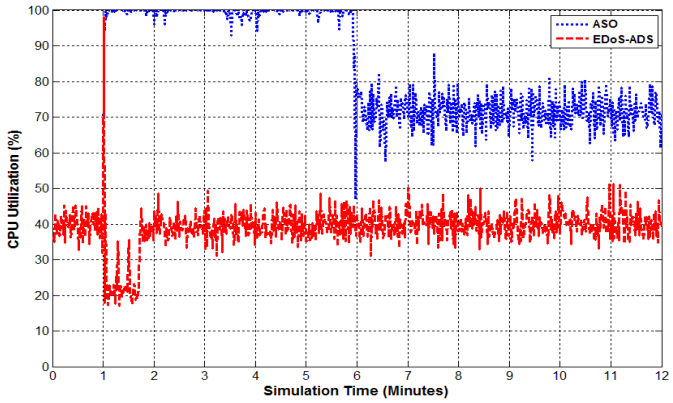

Fig. 14. CPU utilization (same NAT-based network).

does not immediately reach the cloud. Hence, the CPU utilization drops to about 20% which is associated with the CPU usage for serving only the first group requests. The two spikes that appear between minutes one and two are due to the second group legitimate traffic that is forwarded to the cloud after solving the GTT. After minute two, the CPU utilization returns to 40% as the second group clients request rate is allowed to reach to five requests per second. According to (1), a higher request rate is allowed for the second group clients because of the improvement of their TF as a result of solving the GTT.

The legitimate requests response time is shown in Fig. 15. The average EDoS-ADS and ASO response time is 12 milliseconds during the first minute of simulation. Once the attack starts, ASO response time increases to 1.85 seconds due to the queuing delays caused by the attack traffic, and the delay associated with allocating more VM instances to handle that traffic. After six minutes of simulation, ASO response time drops to 24 milliseconds as there are enough VM instances allocated to serve both legitimate and attack traffic. By contrast, when the attack starts, EDoS-ADS uses the **Attack Shell** to block all attack traffic. Hence, the average EDoS-ADS response time remains fixed at 12 milliseconds. Thus, EDoS-ADS successfully eliminates the EDoS attack effect on the legitimate clients response time. Note that the three response time spikes that show between minutes one and two in Fig. 15 are due to the overhead of solving the GTT by the second group clients.

Fig. 16 shows the legitimate requests throughput. The EDoS-ADS and ASO results are identical during the first minute of simulation as the allocated VM instances can

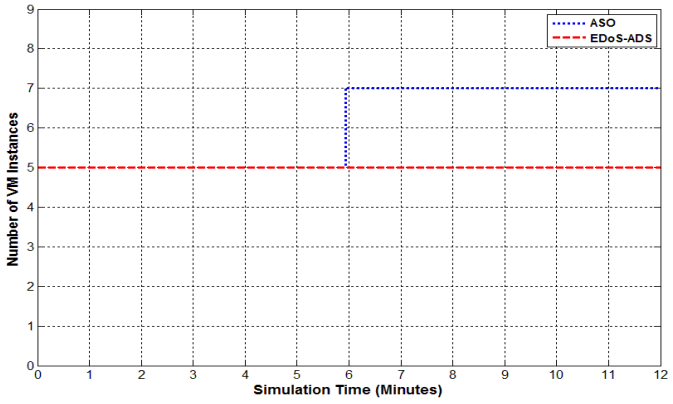

Fig. 13. Number of allocated VMs (same NAT-based network).

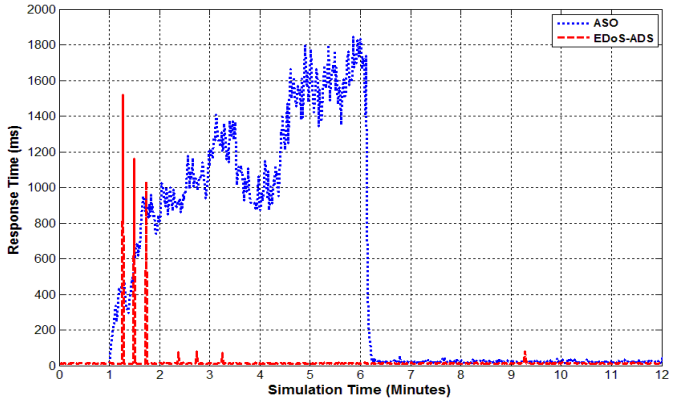

Fig. 15. Response time (same NAT-based network).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2018.2805907, IEEE Transactions on Cloud Computing

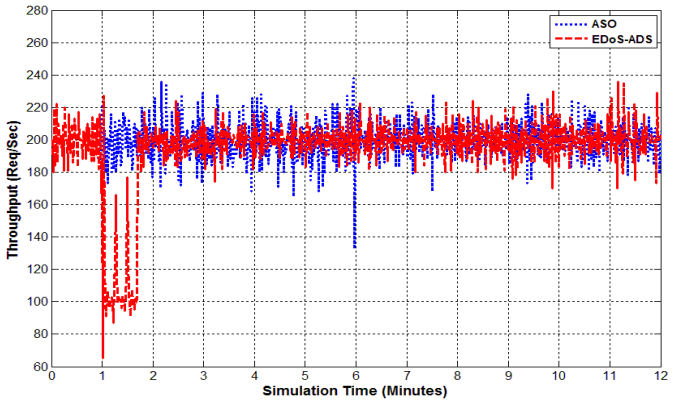

Fig. 16. Legitimate requests throughput (same NAT-based network).

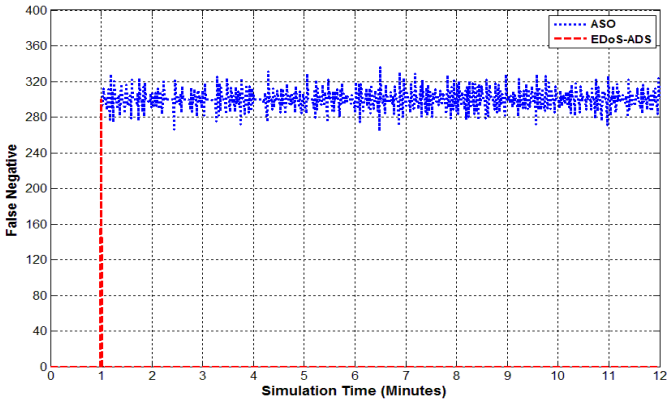

Fig. 17. False negative (same NAT-based network).

handle the received traffic. Further, the average ASO legitimate requests throughput is 200 requests per second during the entire simulation as ASO auto scales to handle both legitimate and attack traffic. Conversely, when the attack starts, EDoS-ADS legitimate requests throughput drops momentarily to 65 requests per second as the LB starts redirecting the incoming requests to the **Attack Shell**. The brief throughput drop is due to the small overhead of the URL redirection requests sent to the legitimate clients, and the delay associated with the response to the GTT by the second legitimate group clients. During the attack that occurs between minutes one and two, the throughput returns to 100 requests per second. Such throughput is associated with the first legitimate group requests only as the second group clients requests are not forwarded to the cloud until such clients respond to the GTT sent to them. Starting from minute two of the simulation, the throughput reaches 200 requests per second. The increase in the throughput is due to an increase in the second group clients request rate as such clients have improved their *Allowable-RPS* by solving the GTT. Thus, the EDoS-ADS legitimate requests throughput is unaffected by the EDoS attack. Further, since EDoS-ADS legitimate requests throughput is equal to the legitimate clients requests rate, then EDoS-ADS is successful in delivering all legitimate requests to the cloud.

Fig. 17 shows the false negative evaluation by counting the number of attack requests per second that succeed in accessing the cloud. The attack requests are generated by clients of groups 3, 4, and 5 with a total rate of 300 requests per second. Such attack requests attempt to access the cloud once the attack begins at minute one. Fig. 17 clearly shows that EDoS-ADS succeeds in immediately blocking all attack requests since the **Attack Shell** enforces the use of the cloud server virtual IP address as stated in subsection 3.2. By contrast, ASO forwards all attack requests to the cloud as ASO does not use an EDoS attack mitigation technique.

To summarize the results of this simulation scenario, EDoS-ADS successfully forwards all legitimate requests to the cloud, and blocks all attack requests. EDoS-ADS does so while using the least number of VM instances needed to handle only legitimate requests. Furthermore, EDoS-ADS ensures that the CPU utilization, and the legitimate requests response time and throughput are unaffected by the attack requests. Finally, EDoS-ADS effectively distinguishes between legitimate clients and attackers even when both belong to the same NAT-based network.

5.3 EDoS-ADS Simulation Results Implications

Based on the simulation results presented in subsections 5.1 and 5.2, it is anticipated that by deploying the EDoS-ADS in a real public cloud that the following benefits will be observed by the cloud clients, adopters, and providers. Note that these observed benefits are limited by the fact that they are based on the results of a simulation as opposed to the results of an actual deployment in a real cloud.

Cloud users' benefits: As evident from Fig. 5 and Fig. 6, the cloud users will notice no degradation in either the response time or the throughput during a flash overcrowd situation. Such a situation can, for example, occur during a sporting or a conference event. More importantly, Fig. 10, 11, 15, and 16 show that the cloud users' response time and throughput will not be affected during an EDoS attack situation. By comparison, consider the case when the cloud employs either no EDoS mitigation technique or employs the EDoS-Shield. Under such a case, it is clear from Fig. 15 that the response time for the cloud users who subscribe to a NAT-based cloud adopter will significantly increase for an extended period when attackers who subscribe to the same NAT-based cloud adopter launch an EDoS attack. Note that, as explained in subsection 5.2.2, the EDoS-Shield (whitelist case) response time results are not shown in Fig. 15 as they resemble the results of not using any mitigation technique, and the response time for the EDoS-Shield (blacklist case) will be zero as both legitimate and attack traffic will be dropped.

Cloud adopters' benefits: The main concern for the cloud adopters is that they do not wish to pay additional cost to the cloud provider for EDoS attack traffic. At the same time, the cloud adopters wish to have no degradation in the response time and throughput for their clients during an EDoS attack. The latter concern has been addressed by the EDoS-ADS technique as explained in the **cloud users' benefits** above. With respect to the cost concern, it is obvious from Fig. 7 that there is no additional cost that is expected during a flash overcrowd situation with the deployment of the EDoS-ADS technique. More notably, the EDoS-ADS technique is expected to significantly benefit the cloud adopters with respect to cost during an EDoS attack situation. Specifically, it is apparent from Fig. 12 that the cost remains constant with the use of EDoS-ADS as opposed to

the cases of either not using an EDoS mitigation technique or using the EDoS-Shield. Thus, the EDoS-ADS effectively addresses the main concern of the cloud adopters about cost.

Cloud providers' benefits: The cloud providers can financially benefit from deploying EDoS-ADS by either selling EDoS-ADS as a feature that the cloud adopters can utilize to reduce their costs, or to meet the cloud providers' service level agreement legal obligations towards their cloud adopters. More importantly, a cloud provider must ensure that whichever EDoS mitigation technique is deployed it does not inadvertently block the entire cloud adopter traffic. This may occur if the traffic is received through the cloud adopter's NAT gateway router and the traffic is generated by a mixture of legitimate clients and attackers who subscribe to the same NAT-based cloud adopter. Subsequently, the EDoS-ADS has demonstrated its ability to distinguish between legitimate and attack traffic that is received from a NAT-based cloud adopter as seen from Fig. 17.

6 Conclusion

The EDoS attack is one of the major threats to the cloud. This paper presents the EDoS Attack Defense Shell (EDoS-ADS) reactive mitigation technique. EDoS-ADS is only triggered when there is suspicious traffic arriving at the cloud. Thus, all incoming traffic are directed to the EDoS-ADS to investigate its legitimacy. Once an EDoS attack is detected, the EDoS-ADS triggers a checking component to differentiate between legitimate users and attackers. Subsequently, attacker requests are dropped while legitimate user requests are directed to the cloud. The EDoS-ADS is the first known technique that is able to effectively identify the legitimacy of clients behind a NAT and prevent the blocking of an entire NAT-based network from accessing the cloud. The EDoS-ADS effectiveness is evaluated using CloudSim simulator. Further, the EDoS-ADS is compared with the EDoS-Shield. The comparison results show that the EDoS-ADS is better than the EDoS-Shield in terms of performance metrics.

Acknowledgments

The authors would like to thank King Fahd University of Petroleum and Minerals for supporting this research and providing the computing facilities.

References

[1] P. Mell, T. Grance *et al.*, "The nist definition of cloud computing," 2011.

[2] F. Gens, "New idc it cloud services survey: Top benefits and challenges," *IDC exchange*, pp. 17–19, 2009.

[3] Y. K. Sinjilawi, M. Q. Al-Nabhan, and E. A. Abu-Shanab, "Addressing security and privacy issues in cloud computing," *Journal of Emerging Technologies in Web Intelligence*, vol. 6, no. 2, pp. 192–199, 2014.

[4] O. Osanaiye, K.-K. R. Choo, and M. Dlodlo, "Distributed denial of service (ddos) resilience in cloud: review and conceptual cloud ddos mitigation framework," *Journal of Network and Computer Applications*, vol. 67, pp. 147–165, 2016.

[5] M. H. Sqalli, F. Al-Haidari, and K. Salah, "Edos-shield-a two-steps mitigation technique against edos attacks in cloud computing," in *Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on*. IEEE, 2011, pp. 49–56.

[6] H. Wang, Z. Xi, F. Li, and S. Chen, "Abusing public third-party services for edos attacks," in *10th USENIX Workshop on Offensive Technologies (WOOT 16)*. USENIX Association, 2016.

[7] N. V. Juliadotter and K.-K. R. Choo, "Cloud attack and risk assessment taxonomy," *IEEE Cloud Computing*, vol. 2, no. 1, pp. 14–20, 2015.

[8] S. Iqbal, M. L. M. Kiah, B. Dhaghighi, M. Hussain, S. Khan, M. K. Khan, and K.-K. R. Choo, "On cloud security attacks: A taxonomy and intrusion detection and prevention as a service," *Journal of Network and Computer Applications*, vol. 74, pp. 98–120, 2016.

[9] K. S. Tep, B. Martini, R. Hunt, and K.-K. R. Choo, "A taxonomy of cloud attack consequences and mitigation strategies: The role of access control and privileged access management," in *Trustcom/BigDataSE/ISPA, 2015 IEEE*, vol. 1. IEEE, 2015, pp. 1073–1080.

[10] N. C. S. N. Iyengar, G. Ganapathy, P. Mogan Kumar, and A. Abraham, "A multilevel thrust filtration defending mechanism against ddos attacks in cloud computing environment," *International Journal of Grid and Utility Computing*, vol. 5, no. 4, pp. 236–248, 2014.

[11] A. K. Marnerides, P. Spachos, P. Chatzimisios, and A. U. Mauthe, "Malware detection in the cloud under ensemble empirical mode decomposition," in *Computing, Networking and Communications (ICNC), 2015 International Conference on*. IEEE, 2015, pp. 82–88.

[12] T. Karnwal, S. Thandapanii, and A. Gnanasekaran, "A filter tree approach to protect cloud computing against xml ddos and http ddos attack," in *Intelligent Informatics*. Springer, 2013, pp. 459–469.

[13] S. Gupta and P. Kumar, "Vm profile based optimized network attack pattern detection scheme for ddos attacks in cloud," in *International Symposium on Security in Computing and Communication*. Springer, 2013, pp. 255–261.

[14] O. Osanaiye, H. Cai, K.-K. R. Choo, A. Dehghantanha, Z. Xu, and M. Dlodlo, "Ensemble-based multi-filter feature selection method for ddos detection in cloud computing," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, p. 130, 2016.

[15] B. Gupta and O. P. Badve, "Taxonomy of dos and ddos attacks and desirable defense mechanism in a cloud computing environment," *Neural Computing and Applications*, pp. 1–28, 2016.

[16] G. Somani, M. S. Gaur, D. Sanghi, M. Conti, and R. Buyya, "Ddos attacks in cloud computing: issues, taxonomy, and future directions," *Computer Communications*, 2017.

[17] N. Agrawal and S. Tapaswi, "Defense schemes for variants of distributed denial-of-service (ddos) attacks in cloud computing: A survey," *Information Security Journal: A Global Perspective*, vol. 26, no. 2, pp. 61–73, 2017.

[18] A. S. Bhingarkar and B. D. Shah, "A survey: Securing cloud infrastructure against edos attack," in *Proceedings of the International Conference on Grid Computing and Applications (GCA)*. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2015, p. 16.

[19] O. Osanaiye, K.-K. R. Choo, and M. Dlodlo, "Change-point cloud ddos detection using packet inter-arrival time," in *Computer Science and Electronic Engineering (CEEC), 2016 8th*. IEEE, 2016, pp. 204–209.

[20] A. Somasundaram, "Economic denial of sustainability attack on cloud-a survey." *ICTACT Journal on Communication Technology*, vol. 7, no. 4, 2016.

[21] A. CloudWatch, "Amazon cloudwatch," 2014.

[22] S. H. Khor and A. Nakao, "spow: On-demand cloud-based eddos mitigation mechanism," in *HotDep (Fifth Workshop on Hot Topics in System Dependability)*, 2009.

[23] V. D. Gligor, "Guaranteeing access in spite of distributed service-flooding attacks," in *International Workshop on Security Protocols*. Springer, 2003, pp. 80–96.

[24] M. N. Kumar, R. Korra, P. Sujatha, and M. Kumar, "Mitigation of economic distributed denial of sustainability (eddos) in cloud computing," in *Proc. of the Intl'Conf. on Advances in Engineering and Technology*, 2011.

[25] M. N. Kumar, P. Sujatha, V. Kalva, R. Nagori, A. K. Katukojwala, and M. Kumar, "Mitigating economic denial of sustainability (edos) in cloud computing using in-cloud scrubber service," in *Computational Intelligence and Communication Networks (CICN), 2012 Fourth International Conference on*. IEEE, 2012, pp. 535–539.

[26] W. Alosaimi and K. Al-Begain, "A new method to mitigate the impacts of the economical denial of sustainability attacks against the cloud," in *Proceedings of the 14th Annual Post Graduates Symposium on the convergence of Telecommunication, Networking and Broadcasting (PGNet)*, 2013, pp. 116–121.

[27] F. Al-Haidari, M. Sqalli, and K. Salah, "Evaluation of the impact

of edos attacks against cloud computing services," *Arabian Journal for Science and Engineering*, vol. 40, no. 3, pp. 773–785, 2015.

[28] W. G. Morein, A. Stavrou, D. L. Cook, A. D. Keromytis, V. Misra, and D. Rubenstein, "Using graphic turing tests to counter automated ddos attacks against web servers," in *Proceedings of the 10th ACM conference on Computer and communications security*. ACM, 2003, pp. 8–19.

[29] F. Al-Haidari, M. H. Sqalli, and K. Salah, "Enhanced edos-shield for mitigating edos attacks originating from spoofed ip addresses," in *Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on*. IEEE, 2012, pp. 1167–1174.

[30] A. Yaar, A. Perrig, and D. Song, "Pi: A path identification mechanism to defend against ddos attacks," in *Security and Privacy, 2003. Proceedings. 2003 Symposium on*. IEEE, 2003, pp. 93–107.

[31] M. Masood, Z. Anwar, S. A. Raza, and M. A. Hur, "Edos armor: a cost effective economic denial of sustainability attack mitigation framework for e-commerce applications in cloud environments," in *Multi Topic Conference (INMIC), 2013 16th International*. IEEE, 2013, pp. 37–42.

[32] Z. A. Baig, S. M. Sait, and F. Binbeshr, "Controlled access to cloud resources for mitigating economic denial of sustainability (edos) attacks," *Computer Networks*, vol. 97, pp. 31–47, 2016.

[33] Y. Gokcen, V. A. Foroushani, and A. N. Z. Heywood, "Can we identify nat behavior by analyzing traffic flows?" in *Security and Privacy Workshops (SPW), 2014 IEEE*. IEEE, 2014, pp. 132–139.

[34] H. S. Cho and Y. K. Noh, "System for preventing normal user being blocked in network address translation (nat) based web service and method for controlling the same," Apr. 30 2013, uS Patent 8,434,141.

[35] D. Bellenger, J. Bertram, A. Budina, A. Koschel, B. Pfänder, C. Serowy, I. Astrova, S. Grivas, and M. Schaaf, "Scaling in cloud environments," *Recent Researches in Computer Science*, vol. 33, 2011.

[36] J. Idziorek, "Discrete event simulation model for analysis of horizontal scaling in the cloud computing model," in *Proceedings of the Winter Simulation Conference*. Winter Simulation Conference, 2010, pp. 3004–3014.

[37] Y. Wang, K.-J. Lin, D. S. Wong, and V. Varadharajan, "The design of a rule-based and event-driven trust management framework," in *e-Business Engineering, 2007. ICEBE 2007. IEEE International Conference on*. IEEE, 2007, pp. 97–104.

[38] Z. Zhou, Y. Luo, L. Guo, and L. Sun, "Assessment of p2p trust model based on fuzzy comprehensive evaluation." *JSW*, vol. 8, no. 11, pp. 2711–2714, 2013.

[39] L. Von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum, "recaptcha: Human-based character recognition via web security measures," *Science*, vol. 321, no. 5895, pp. 1465–1468, 2008.

[40] K. Bhargrava, D. Brewer, and K. Li, "A study of url redirection indicating spam," in *Proceeding Conference on Email and Anti-Spam*. Citeseer, 2009.

[41] R. Fielding and J. Reschke, "Hypertext transfer protocol (http/1.1): Semantics and content," 2014.

[42] S. Islam, K. Lee, A. Fekete, and A. Liu, "How a consumer can measure elasticity for cloud platforms," in *Proceedings of the 3rd ACM/SPEC International Conference on Performance Engineering*. ACM, 2012, pp. 85–96.

[43] H. Wu, A. N. Tantawi, and T. Yu, "A self-optimizing workload management solution for cloud applications," in *Web Services (ICWS), 2013 IEEE 20th International Conference on*. IEEE, 2013, pp. 483–490.

[44] D. Moore, C. Shannon, D. J. Brown, G. M. Voelker, and S. Savage, "Inferring internet denial-of-service activity," *ACM Transactions on Computer Systems (TOCS)*, vol. 24, no. 2, pp. 115–139, 2006.

[45] X. Bao, H. Hong, Z. Zhou, Y. Liu, and Z. Cao, "Nsfocus ddos threat report 2013," *NSFOCUS Information Technology*, 2013.

[46] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose, and R. Buyya, "Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and experience*, vol. 41, no. 1, pp. 23–50, 2011.

[47] A. Shawahna. EDoS-ADS. GitHub, GitHub repository. Available: https://github.com/Ahmad-Shawahna/EDoS-ADS.git, 2017.

[48] M. F. Arlitt and C. L. Williamson, "Internet web servers: Workload characterization and performance implications," *IEEE/ACM Transactions on networking*, vol. 5, no. 5, pp. 631–645, 1997.

[49] K. Xiong and H. Perros, "Service performance and analysis in cloud computing," in *Services-I, 2009 World Conference on*. IEEE, 2009, pp. 693–700.

[50] X. Shen, H. Chen, J. Dai, and W. Dai, "The finite element method for computing the stationary distribution of an srbm in a hypercube with applications to finite buffer queueing networks," *Queueing Systems*, vol. 42, no. 1, pp. 33–62, 2002.

[51] Amazon. Amazon EC2 Pricing [Online]. Available: https://aws.amazon.com/ec2/pricing/on-demand/, 2017.

[52] A. Shawahna, "EDoS Attack Defense Shell (EDoS-ADS): An Enhanced Mitigation Technique Against Economic Denial of Sustainability (EDoS) Attacks for Controlling the Access to Cloud Resources," Master's thesis, King Fahd University of Petroleum and Minerals, Saudi Arabia, 2016.

[53] S. Alsowail, M. H. Sqalli, M. Abu-Amara, Z. Baig, and K. Salah, "An experimental evaluation of the edos-shield mitigation technique for securing the cloud," *Arabian Journal for Science and Engineering*, vol. 41, no. 12, pp. 5037–5047, 2016.

[54] D. Catteddu, "Cloud computing: benefits, risks and recommendations for information security," in *Web application security*. Springer, 2010, pp. 17–17.

[55] K. Claffy, G. Miller, and K. Thompson, "The nature of the beast: Recent traffic measurements from an internet backbone," in *Proceedings of INET*, vol. 98, 1998, pp. 21–24.

[56] H. Liu and S. Wee, "Web server farm in the cloud: Performance evaluation and dynamic architecture," in *IEEE International Conference on Cloud Computing*. Springer, 2009, pp. 369–380.

[57] Website Optimization, LLC. Beware HTTP Redirects for SEO and Performance [Online]. Available: http://www.websiteoptimization.com/speed/tweak/redirect/, 2017.

[58] C. Reiss, A. Tumanov, G. R. Ganger, R. H. Katz, and M. A. Kozuch, "Towards understanding heterogeneous clouds at scale: Google trace analysis," *Intel Science and Technology Center for Cloud Computing, Tech. Rep*, p. 84, 2012.

Ahmad Shawahna obtained M.S. in computer engineering from King Fahd University of Petroleum and Minerals (KFUPM), Saudi Arabia, in 2016. He also received the B.Sc degree in computer engineering from An-Najah National University, Palestine, in 2012. He is currently working at CCITR, KFUPM. His research interests include wireless security, network security, Internet of Things (IoT), and cloud computing.

Marwan Abu-Amara received the M.S. and Ph.D. degrees in electrical and computer engineering from Texas A&M University, USA, in 1991 and 1995, respectively. From 1995 to 2003, he worked for Nortel Networks, USA, as a senior technical advisor. Since 2003, he has been with the Computer Engineering Department, KFUPM, Saudi Arabia. His research interests include cloud and Internet security and resiliency, IoT, and wireless communications.

Ashraf S. H. Mahmoud received the Ph.D. degree in electrical engineering from Carleton University, Canada, in 1997. From 1997 to 2002, he was with Nortel Networks, Canada, as a Senior Radio Systems Engineer. Since 2002, he has been with the Computer Engineering Department, KFUPM, Saudi Arabia. His research interests include performance evaluation, simulation and modeling, and software defined networking.

Yahya Osais is a faculty in the department of computer engineering at King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia. He obtained his B.S. and M.S. from the same department in 2000 and 2003, respectively. In 2010, he obtained his Ph.D. from Carleton University, Canada. His current research interests include stochastic modeling and simulation, cyber-physical systems, and IoT.