BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **13**, No 1

Sofia • 2013

Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2013-0005

Escape-on-Sight: An Efficient and Scalable Mechanism for Escaping DDoS Attacks in Cloud Computing Environment

N. Jeyanthi, N. Ch. S. N. Iyengar***

[*] *School of Information Technology and Engineering, VIT University, Vellore-632014, Tamilnadu, India*
[**] *School of Computing Science and Engineering, VIT University, Vellore-632014, Tamilnadu, India*
Emails: njeyanthi@vit.ac.in; nchsniyengar48@gmail.com

Abstract: *Availability is one of the primary security issues in Cloud computing environment. The existing solutions that address the availability related issues can be applied in cloud computing environment, but because of their unique characteristics, such as on-demand self service, rapid elasticity, etc., there is a need to develop a detection mechanism that must satisfy the characteristics and an optimal profit for the Cloud Service Provider* (*CSP*). *A solution named Escape-on-Sight* (*EoS*) *algorithm is proposed in this paper that helps in detecting the attacker's characteristics by analyzing traffic conditions stage by stage and protects the Data Center* (*DC*) *from malicious traffic. The profit analysis shows that the proposed approach has a reasonable chance of deploying EoS mechanism at DCs that are prone to DDoS attacks.*

Keywords: *Availability; Cloud computing; datacenter; DDoS; EoS; CSP.*

1. Introduction

Cloud computing technology resides in the category of high end computing. In order to provide various cloud based services using this technology, CSPs(Cloud Service Providers) build and manage large Data Centres (DC) consisting of high capacity storage resources, hardware resources, memory and network resources. Clients access these services via a web browser. In order to ensure high availability of the offered services, the DC resources must be protected from DDoS attack threats.

Distributed Denial of Service (DDoS) is one of the serious security threats that challenge the availability of the DC resources to the intended clients. The existing solutions to monitor the incoming traffic to detect the DDoS attacks become ineffective if the attackers' traffic intensity is high. Therefore it is necessary to devise schemes that will detect the DDoS attacks even when the traffic intensity is high and to deactivate DDoS attackers, in order to serve the legitimate users with DC resources. With DDoS attack, an attempt of identifying the source is almost impossible as the huge tries to compromise the DC.

In this paper, a new algorithm called Escape-on-Sight (EoS) is proposed to detect and deactivate the several DDoS attack types at various stages. This algorithm detects the overload threat and instantly locks the port and disallows the attacker entry, but serves the legitimate client who follows the legitimate profile. The organization of the paper is as follows: Section 2 presents a problem definition and related work. Section 3 presents an overview of the proposed model. Section 4 elaborates the working mechanism of the proposed approach. Section 5 shows the design of the proposed approach. Section 6 describes the modelling procedure of EoS. Section 7 lists the performance evaluation. Section 8 presents advantage and profit analysis of EoS and finally Section 9 concludes the work.

2. Existing work on DDos

There are several security issues that affect the performance of cloud computing service efficiency. Various mechanisms used for DDoS detection and DDoS scenarios and their after-effects in real-time are described in [1]. Neural classifier [2] architecture has four stages for detecting DDoS attacks. They are Data collection, pre-processing, classification, response. The attacks are classified as true positive, true negative, false positive and false negative, and this improves the detection accuracy. Migration based response [3] is a method based on distributed auctioneers and bidders for defending the DoS attacks. But when the attackers learn the auctioneers' address, they launch the flooding attack. Security issues [4] that arise from the service provider and external attackers are compromised through public key cryptography.

Intrusion Detection System (IDS) [5] enhances the system by distributing the IDS nodes across the network. Host IDS collect audit data from the operating system. Network IDS collect data from network packets. When any malicious

intrusion is detected, the system generates reports and alerts. Fault tolerant workflow scheduling [6] makes use of failure probability information. Combining the heuristic information and replicating the tasks helps meeting the task deadline and saves resources. Anomaly Detection System [7] creates a baseline profile; when any deviation is found, the threat is detected. It clearly distinguishes the attacker behavior and detects the unknown attacks. This scheme needs a training phase, and when inappropriate dataset is recorded at training phase, this may lead to poor detection accuracy. A fault tolerant mechanism [8] in cloud computing improves the availability of DC by creating a checkpoint replication at each node. Intrusion Detection System [9] explains the defend mechanism of TCP flooding and the virtual switch allows only the traffic based on pre-defined rules, and in-bound and out-bound traffic restricts the DDoS attackers entry.

Security threats in cloud computing [13] are discussed and the possible solutions are addressed. The cloud service provider [14, 15] should be able to provide the intended services and be able to manage the security from serious threats such as reliability, availability and security. The main difference between the legitimate request and the DDoS attacker request is analyzed using varied traffic pattern [10]. DDoS attackers employ botnets to launch DDoS to deplete server resources. This can be detected in real time web browsers by employing the CAPTCHA [11], which requires human user's knowledge to solve a simple puzzle. Fuzzy Pattern Recognition Filtering [12] mechanism for Botnet has three stages, namely traffic reduction, feature extraction, fuzzy pattern recognition.

The DDoS launch against cloud computing DC leads to two problems: revenue loss and loss of fame for CSP and unnecessary usage charge for clients. The motivation of this paper is to propose a server end solution, without much overhead at DC. Hence novel partial/delegated detection architecture at the server end is proposed to respond quickly and to offer efficient service to legitimate clients.

3. Overview of the proposed model

3.1. System architecture

The architecture of the proposed system is shown in Fig. 1. DC requesters can be either a legitimate client or an attacker or combination of both. Whenever the requester needs to communicate to DC, the requester will be validated based on the behavior, i.e., the traffic characteristics are periodically evaluated. The black-bordered rectangular area represents the activities that are carried out at the server end. The proposed scheme is not a server-side detection mechanism, it is rather a partial/delegated server-side DDoS prevention mechanism, because each component has its own functionality in detecting the flooding attack type. So, any flooding threat is detected, as they are mitigated by filtering the attackers' requests at firewall before reaching DC.

3.2. Rationale for Escape-on-Sight

The **firewall** prevents the misbehaving requesters' entry into the server end. The **Traffic Analyzer** continuously monitors the incoming traffic and alerts whenever the current traffic load exceeds the link capacity (usually by abnormal traffic initiation threats). The **Router** directs the incoming packet to the load balancer, only when the consequent packets follow Normal Condition in a Matchboard Profiler. The **Load Balancer** is configured to bypass only the compatible packets and the packets arriving to service applications. **VM Router Switch** helps in maintaining the virtual machines in each physical host of each Data Center. This supports hierarchical load balancing by balancing the load at VM (Virtual Machine) level and maintains the requesters' inter-arrival time of each packet. When any deviation in a legitimate pattern is found, the packets are forwarded to the **Packet Analyzer** that extracts the header information and passes to the firewall to prevent the packet entry from unauthorized requester until the session expires. The packet is now destroyed and further transmission of packets is denied for the unauthorized requester. The **Data Centers** provide resources only to legitimate clients, and not to aggressive legitimate clients.

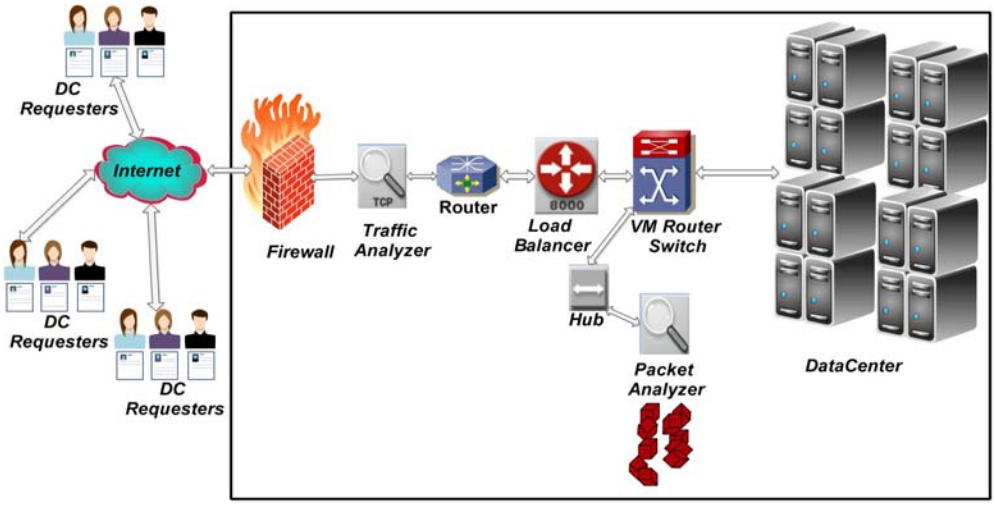


Fig. 1. Architecture of the proposed system

4. Working mechanism of EoS

In this section, first the pseudo-code that explains the working mechanism has been addressed. The various possible DDoS attack scenarios are analyzed in the later part.

4.1. Pseudo code of Escape-on-Sight

```
If (Requesters needs to access DC resources)
{
    At firewall:
    If (Requester IP is not blocked)
    {
        At Traffic Analyzer:
        If (Current traffic <= Link Capacity)
        {
            If (Matchboard profilers == Normalcy)
            {
                Route Packets towards Load Balancer.
                At Load Balancer:
                Based on the server load forward the Legitimate traffic to particular DC.
                At Router Switch:
                If (No ACK || Sequence Number mismatch)
                {
                  Expire the client session at server end. // Spoof Attack Detection.
                }
                 If (for each requester [Packet Inter-Arrival time < Legitimate Pattern])
                 {
                    Forward the packets to packet Analyzer.
                    At Packet Analyzer:
                    Packet header extracted and the requester IP forwarded to firewall filtration.
                    Packet is destroyed.  // Aggressive Legitimate users Detection.
                }
                If (for each requester [Packet Inter-Arrival time >= Legitimate Pattern] && overload)
                {
                  Traffic is caused by huge simultaneous legitimates. // Flash Crowd Detection.
                }
                Else
                {
                     Requesters does not involve in any overload condition and allowed to access DC
                    resources // Legitimate user is serviced.
                }
            Else
            {
                At firewall:
                Filter the packets and further packets from the particular Requester. // DDoS Detection.
             }
        Else
        {
            Buffer the packets overloaded and forward with acceptable delay to the Matchboard Profilers.
        }
    Else
    {
          Filter the requester packet entry into the network.
     }
}
```

Fig. 2. Pseudo code of EoS

4.2. DDoS attack scenario analyses

- Flooding by attackers – DDoS is flooding by malicious/incompatible packets by the attackers towards the DataCenter. This kind of overload threat could be easily detected by a Matchboard Profiler. If the attacker characteristic is found, then the user could be filtered by the firewall.

- Flooding by legitimates (Flash Crowd) – Flash Crowd is an overload condition that is caused by the legitimate users, where huge numbers of legitimate users request the DC resources simultaneously. This can be solved by buffering the excess number of requests that makes this overload condition remain alive only for a certain period of time.
- Flooding by spoofing attackers – caused by impersonation that can be detected by acknowledging each request and by maintaining the sequence number of the requests and requesters' IP (Internet Protocol) address.
- Flooding by aggressive legitimates – caused by aggressive users, it is an overload condition where the legitimate users flood the server with the requests that slow down the performance of DC. This condition is critical to detect, because the overload has legitimate characteristics. By maintaining the nter-arrival time of users' packets by a back-off timer, this attack can be detected.

5. Design of EoS algorithm

Detailed design of the proposed EoS mechanism is elaborated below.

5.1. Traffic analysis

Whenever the requester sends a request for DC resource access, the first step is to direct the requests to the traffic analyzer. When the incoming traffic exceeds the link capacity, the abnormal traffic is detected and it is passed onto customized Routers.

5.2. Attack detection

Once the request packets bypass the customized router, it assures that the packets are legitimates. The job of the customized router is to compare the packet arrival rate of each incoming IP with the Matchboard Profiler. If the packet arrival rate exceeds the nominal profile, then the attacker is detected and is blocked through the firewall based on his IP. Otherwise the traffic is redirected to the load balancers, which route the packets based on the load balancing policy (server load, round-robin and failure recovery).

5.3. Attack classification

The load balancer directs the request packets to the intended DC. Before the request packets reach DC, they are parsed by a VM Router Switch. This connects the VMs of each physical host of DC.

Attack traffic – when the incoming packets at VM router switch result in ACK ahead of a sequence number or variation at TTL, the spoofed attacker is detected and added to the firewall filtration.

Legitimate traffic – the legitimate traffic from usually aggressive clients are hard to detect because they follow legitimacy, but try to overload the DC. The flash crowd is a legitimate traffic where several users access simultaneously for DC

resource. The difference between the aggressive clients and flash crowd is that the aggressive clients are identified through packet inter-arrival rate.

By this mechanism, only legitimate clients would be allowed to access DC for that particular session.

5.4. Attack prevention

The firewall filters the IP that are instructed by the Matchboard Profiler Router. This acts as a preventive measure and any attempt to access DC will never be allowed until the session is updated once the attacker is logged out of session. Otherwise, the attackers are blocked.

6. Modelling EoS algorithm

To evaluate the performance of our EoS algorithm, a customized world map scenario is created in OPNET simulator. The attack scenarios reflect the DDoS attack launched by sophisticated DDoS tools like Low Orbit Ion cannon [16]. The OPNET cloud readiness [17, 18] explains more about cloud computing in OPNET. The proposed approach is assessed for end-to-end response time [18]. OPNET supports simulation of DDoS and performance comparison for QoS (Quality-of-Service) Application in On-Demand Cloud Computing [19, 20]. The distributed DCs are created and configured (Asia, Africa, Australia, and South America), each one with 5 physical hosts and 160 VMs with TIME_SHARED multi-tasking capability. EoS Algorithm is tested with three different applications (Remote login, email, HTTP) to check different sizes of data. In addition, we have also deployed 1000 legitimate clients and 300 attackers distributed around the globe. Asia is assumed as a victim DC to suffer DDoS attack from distributed attackers.

7. Performance evaluation

The performance evaluation shown in Figs 4-7 has three scenarios. They are namely: Scenario a (simulation of network traffic only with legitimate requests), Scenario b (simulation of DDoS attack) and Scenario c (deployment of Escape-on-Sight algorithm under DDoS attack).

7.1. Attacker strength towards a Victim DC

The traffic rate is the average number of packets forwarded per second to the email application, Remote login application, and HTTP application to each DC. The Flooding Traffic Rate, generated by distributed attackers is identified towards the victim DataCenter, ASIA DC.

7.2. Request load at DC

The request load is the rate at which Email requests, Remote login requests, HTTP requests arrive at the server. Note that these requests could belong to different

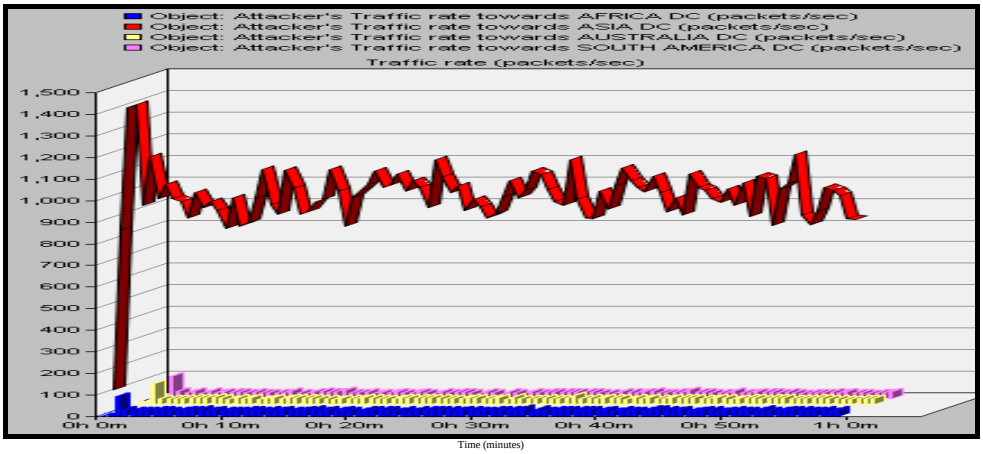
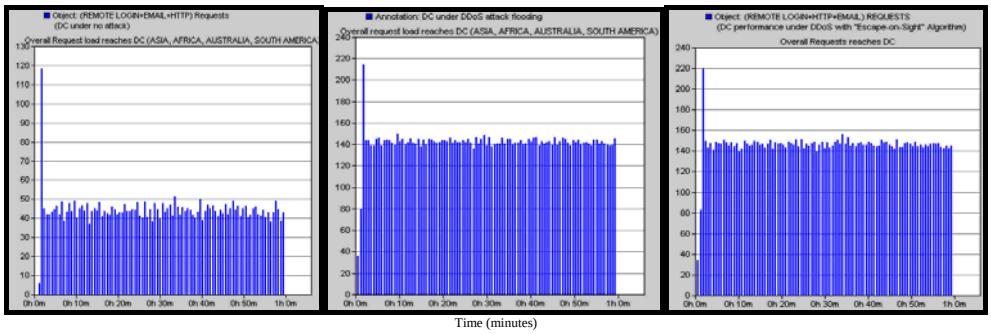


Fig. 3. Traffic rate initiated towards Datacenters



(a) (b) (c)

Fig. 4. Request load at DC: No attack (a); DDoS attack (b); DDoS with EoS (c)

sessions maintained at the server. Also note that the requests for the same session is queued until the first request is completed.

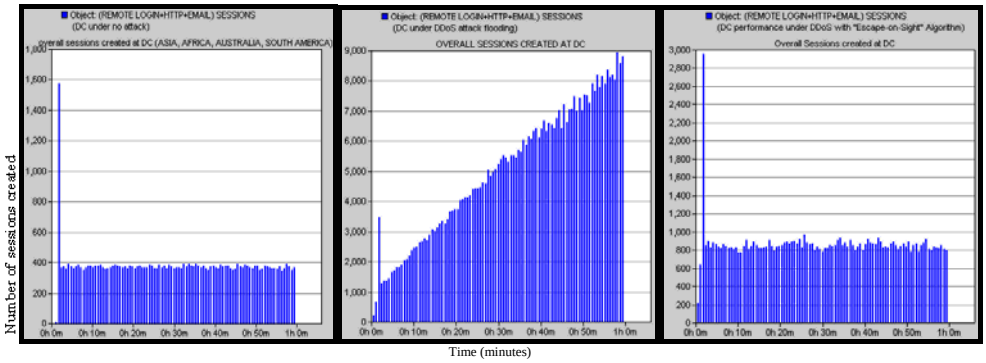
$$(1)\quad \text{Request}_{\text{Load}} = \sum_{i=1}^{N} (\text{RL}_{\text{HTTP}} + \text{RL}_{\text{Email}} + \text{RL}_{\text{Telnet}} + \text{RA}_{\text{HTTP}} + \text{RA}_{\text{Email}} + \text{RA}_{\text{Telnet}}),$$

where $\text{Request}_{\text{Load}}$ is the Request load that reaches DC without any attack traffic, N is the total number of DCs, RL_{HTTP} is the number of legitimate HTTP requests reaching DC, RL_{Email} is the number of legitimate email requests reaching DC, $\text{RL}_{\text{Telnet}}$ is the number of legitimate Telnet requests reaching DC, RA_{HTTP} is Attack HTTP requests reaching DC, which is zero while measuring the request load with no attack, RA_{Email} is the number of the attack Email requests that reaches DC. This attribute is zero while measuring the request load with no attack and $\text{RA}_{\text{Telnet}}$ is the number of Attack Telnet requests reaching DC, which is zero while measuring the request load with no attack.

Fig. 4a shows that initially all the application requesters try to reach DC simultaneously and once the traffic overload is identified, they are controlled by switching the traffic to serial ordered requests to reach DC. Fig. 4b shows the traffic is uncontrolled and the request load shows the evidence of DDoS attack towards DC. Fig. 4c shows the request load that is trying to reach DC, i.e., the traffic at the traffic analyzer.

7.3. Session created at DC

The sessions created at DC represent the current number of email, HTTP, and Telnet sessions on this server. This statistic is intended to provide a picture of how the server is loaded with sessions.



(a) (b) (c)

Fig. 5. Session created at DC: No attack (a); DDoS attack (b); DDoS with EoS (c)

$$(2)\quad \text{Session}_{\text{Load}} = \sum_{i=1}^{N} (\text{SL}_{\text{HTTP}} + \text{SL}_{\text{Email}} + \text{SL}_{\text{Telnet}} + \text{SA}_{\text{HTTP}} + \text{SA}_{\text{Email}} + \text{SA}_{\text{Telnet}}),$$

where $\text{Session}_{\text{Load}}$ is the Request load that reaches DC without any attack traffic, N is the total number of DCs, SL_{HTTP} is the Legitimate HTTP sessions at DC, SL_{Email} is the Legitimate Email sessions at DC, $\text{SL}_{\text{Telnet}}$ is Legitimate Telnet sessions at DC, SA_{HTTP} is Attack HTTP sessions at DC, which is zero while measuring the session load with no attack, SA_{Email} is Attack Email sessions at DC, which is zero while measuring the session load with no attack and $\text{SA}_{\text{Telnet}}$ is the Attack Telnet sessions at DC, which is zero while measuring the session load with no attack. From Fig. 5c it can be observed that the proposed approach would create a session only for legitimate clients, outwitting attackers.

7.4. Remote login response time

Response time is the time elapsed between sending a request and receiving the response for the remote login application (the time taken by a DC to respond to the requests arriving from the requester).

Table 1. Remote login response time

DC location	Minimum time, s			Mean time, s			Maximum time, s		
	No attack	DDoS attack	EoS at DDoS	No attack	DDoS attack	EoS at DDoS	No attack	DDoS attack	EoS at DDoS
Asia	0.06	0.05	0.16	0.08	4.69	0.18	0.09	14.94	0.17
South America	0.09	0.09	0.14	0.11	1.00	0.18	0.12	8.00	0.16

$$\text{Response time (APP)} = \left(\left(\frac{\text{Request size}}{\text{Link capacity}} \right) + S_{\text{prtime}} + \text{delay} \right), \quad (3)$$

where Response time (APP) is the Response time of any application APP, the Request size is the size of data to be transmitted across the network, Link capacity is the Maximum data transmission rate, S_{prtime} is the Server processing time, and *delay* is the Time lag across the network ,which is zero while measuring the response time of an application at no attack. Equation 3 results in Tables 1, 2, 4. The response time for telnet requesters is shown in Table 1. The network with no attacker traffic has quicker response time. The response time under DDoS attack scenario is poor and EoS scenario shows that the Telnet response time is far better than DDoS Scenario. The delay at the victim DC is to segregate the packet header and examine the characteristics at DC end. Once the threat is identified and filtered by the firewall, the delay at victim DC is small.

7.5. Email response time

The email response time is the time elapsed between sending a request for emails and receiving emails from the email server. This time includes the signaling delay for the connection setup.

Table 2. Email response time

DC location	Minimum time, s			Mean time, s			Maximum time, s		
	No attack	DDoS attack	EoS at DDoS	No attack	DDoS attack	EoS at DDoS	No attack	DDoS attack	EoS at DDoS
Africa	0.21	0.21	0.39	0.45	15.88	0.50	0.64	70.75	0.57
Asia	0.29	0.23	0.78	0.45	12.76	0.86	0.71	46.20	0.94
Australia	0.49	0.44	0.92	0.72	7.34	1.01	0.89	26.89	1.09
South America	0.50	0.61	0.84	0.80	5.16	0.92	0.96	19.18	1.00

Table 2 shows that at no attack the legitimate load is balanced and all distributed DC process the application load and respond quickly. DDoS attack scenario shows the poor response time at the victim DC, but the email attackers located in Africa, when try to flood the attack packets at the victim target, are captured and processed by a built-in load balancer and diverted to Africa (the closest) DC thus shows a much poorer response time. EoS scenario shows the email response time is nearly equal at all DC, because the load balancing policy is set based on the server load.

7.6. Total number of HTTP pages downloaded

The total number of the pages downloaded is the count of HTTP page response sent by the DC to the Requester who sends the HTTP page request.

Table 3. Total number of pages downloaded from DC at various scenarios

Scenario	Minimum (No of pages)	Mean (No of pages)	Maximum (No of pages)
1. No attack	0	233.23	577
2. DDoS attack	115	3715.74	4064
3. Escape-on-Sight	0	231.35	580

The number of pages downloaded at various scenarios is shown in Table 3. The attackers are continuously sending HTTP page request, which floods and disallows legitimates to reach DC. After deploying EoS, this flooding is detected and neglected, thereby only the legitimate requests are serviced. This proves that the EoS is highly active in detecting the attack characteristics with the acceptable response times shown in Table 4 (EoS at DDoS).

7.7. HTTP page response time

HTTP page response time is the time required to retrieve the entire page with all the contained inline objects.

Table 4. HTTP page response time

DC location	Minimum time, s			Mean time, s			Maximum time, s		
	No attack	DDoS attack	EoS at DDoS	No attack	DDoS attack	EoS at DDoS	No attack	DDoS attack	EoS at DDoS
Africa	0.41	0.43	0.80	0.57	18.38	0.86	0.62	27.37	0.91
Asia	0.55	0.57	1.02	0.61	15.46	1.07	0.68	24.04	1.15
Australia	0.43	0.44	0.83	0.48	8.19	0.88	0.53	12.70	0.94
South America	0.78	0.91	1.07	0.91	5.50	1.12	0.99	9.65	1.21

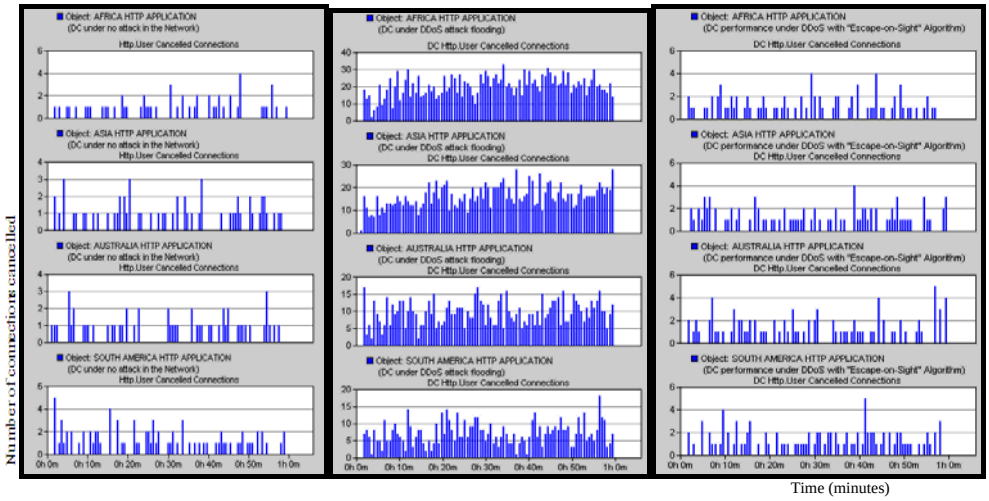
Table 4 shows the Response time of the network behaviour under no attack, the attacker HTTP flood attempt at DC and the response time only for the legitimate clients at EoS scenario. By comparing Table 3, scenario 1 and 3, we could easily predict that the number of pages downloaded is less, so is the response time of HTTP page request.

7.8. User connection cancellation

User connection cancellation is a statistic that represents the number of times a client tries to set up a connection to a given server, though a connection to that server is already open. This represents essentially a page request while downloading

is in progress. For instance, a user clicks on a link in a page that is still downloading.

Fig. 6a shows the collision of the legitimate request while routing. Fig. 6b shows the attackers intend to cancel the connection and to reconnect to DC which requires an additional amount of time for DC to perform. This creates a delay at DC. Fig. 6c shows that EoS approach has better resource reservation.



(a) (b) (c)

Fig. 6. User connection cancellation: No attack (a); DDoS attack (b); DDoS with EoS (c)

7.9. Port based request load

The port based request load is the statistic that represents the average number of packets successfully received by the DC channel per second.

Table 5. Port based request load

Request type	Minimum (requests)			Mean (requests)			Maximum (requests)		
	No attack	DDoS attack	EoS at DDoS	No attack	DDoS attack	EoS at DDoS	No attack	DDoS attack	EoS at DDoS
HTTP (Port 80)	0.0	0.0	0.0	273.48	8146.57	8463.77	22809.88	158082	143772
Email (port 25)	0.0	0.0	0.0	101.07	5067.69	4910.3	8172.88	344452	294469
Telnet (Port 23)	0.0	0.0	0.0	43.99	465.09	418.40	133.44	1209	921

8. Advantages of EoS algorithm

The simulation results in terms of a graph show better performance for our proposed Escape-on-Sight approach under DDoS. The results proved that our approach is suitable to deploy to DC prone to DDoS attack.

8.1. Witnessed advantages

- ***Novel Intermediary Architecture*** – the special hardware is in place to detect and treat the attackers and the cloud DC will only serve legitimates saving time.
- ***Highly sensitive to traffic behavior*** – an efficient traffic analyzer data structure continuously senses the incoming traffic and reports immediately whenever any abnormal traffic is found. The traffic analysis is non-probabilistic to improve the abnormal traffic detection accuracy.
- ***Better response time comparatively*** – the attacker characteristics are detected and filtered earlier to disallow any further transmission and this paves the way to serve legitimate clients quickly.
- ***EoS, sheer virtual level switch*** – almost any kind of DDoS flood can be detected at earlier time in a scalable manner. Also, the proposed approach reduces the channel congestion and offers better response time even at the time of DDoS. When any unabated attack traffic is found, the application port at DC is locked and now DC escapes from DDoS and the data resides in DC is prevented.
- ***Hierarchical Load Balancing*** – load balancers among DC applies "Divide the attacker's traffic and conquer the attacker's traffic". But EoS uses a virtual level VM router switch for balancing the load among VMs resides in a different physical host of each DC. This still offers better response time.

8.2. Profit analysis

In order to simulate the real-time attack scenario, experiments are designed with the characteristics of botnet and distributed attackers. The attackers' group is located around the victim DC and the attack is launched and paused among groups. So, once the attackers are detected early, they are ingress filtered, which saves resources and ultimately improves the performance, revenue and availability.

The total cost incurred, calculated based on the resources used to complete the task, is derived in equation

$$\text{(4)} \quad \text{Total cost incurred at DC} = \sum_{i=1}^{N} (\text{Cost}_{\text{BW}} + \text{Cost}_{\text{MEM}} + \text{Cost}_{\text{VM}} + \text{Cost}_{\text{DS}}),$$

where N is the time in hours, Cost_{BW} is the width cost, Cost_{MEM} is the RAM cost of each physical equipment, Cost_{VM} is the VM cost of each physical equipment, and Cost_{DS} is the cost of the data stored within the DC.

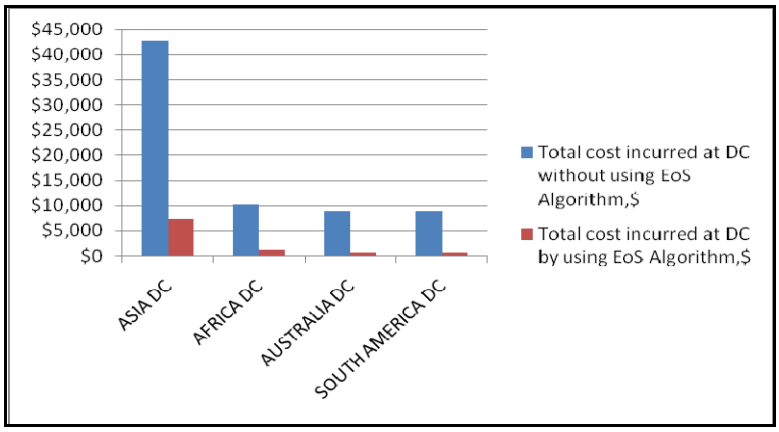


Fig. 7. Profit analysis of EoS

Fig. 7 shows the profit analysis of the experiment scenario. It also shows that the huge cost is incurred at Asia DC, as it is the victim. This top level results in Fig. 7 show that our approach behaves better in detecting DDoS attacks with efficiently improving revenue. The costs used are 0.1 ($/Gb) for any data transmission at DC and 0.05 ($ per 1 s) for any memory resident operations at DC. The extreme difference in a profit is due to the detection of the attacker at their initiation and preventing their subsequent entry towards DC. This paves the way to improve the availability with an acceptable response time shown in Tables 1, 2 , 4.

9. Conclusion and future work

The proposed Escape-on-Sight algorithm helps in identifying the DDoS attackers and also analyzes other causes of overload. This scheme also identifies the aggressive legitimate users and prevents their entering the firewall until the session expiry. This considerably reduces the load at the DC, which is a direct advantage of this approach. These sessions in turn could be used for other legitimate users to improve the performance of the DC by serving legitimates.

The future work is aimed at improving the availability of the DC by resolving other security issues that indirectly improving the DC performance. The simulated results prove that EoS algorithm suits better to DC that is prone to any overload conditions, that are discussed in this work. The improvement in performance is essential, at the same time without destructing the existing protocols. The proposed scheme is efficient in terms of serving the clients by caring the time-sensitiveness, a characteristic of cloud computing. And the scheme is scalable, i.e., the detection capability can be increased with parallelizing the required hardware that is responsible for the improved detection accuracy even at increased traffic. Eventually, the proposed scheme has shown the profit analysis of our approach, which highlights the ingress filtration of the attackers at an earlier stage.

References

1. Janczewski, L. J., D. Reamer, J. Brendel. Handling Distributed Denial-of-Service Attacks. – Information Security Technical Report, Vol. **6**, 2001, 37-44.
2. Kumar, P. A. R., S. Selvakumar. Distributed Denial of Service Attack Detection Using an Ensemble of Neural Classifier. – Computer Communications, Vol. **34**, 2011, 1328-1341.
3. Lent, R. Evaluating a Migration-Based Response to DoS Attacks in a System of Distributed Auctions. – Computers & Amp Security, Vol. **31**, 2012, 327-343.
4. Singh, K., I. Kharbanda, N. Kaur. Security Issues Occur in Cloud Computing and There Solutions. – International Journal on Computer Science and Engineering, Vol. **4**, 2012, 945-949.
5. Huang, C-C, J. Ku. A Cooperative Intrusion Detection System Framework for Cloud Computing Networks. – In: 39th International Conference on Parallel Processing Workshops, 2010, 280-284.
6. Jayadivya, S. K., J. S. Nirmala, M. S. Bhanu. Fault Tolerant Workflow Scheduling Based on Replication and Resubmission of Tasks in Cloud Computing. – International Journal on Computer Science and Engineering, Vol. **4**, 2012, 996-1006.
7. Patcha, A., J.-M. Park. An Overview of Anomaly Detection Techniques: Existing Solutions and Latest Technological Trends, Computer Networks. – The International Journal of Computer and Telecommunications Networking, Vol. **1**, 2007, 3448-3470.
8. Bansal, S., S. Sharma, I. Trivedi, M. Ghosh. Improved Self Fused Check Pointing Replication for Handling Multiple Faults in Cloud Computing. – International Journal on Computer Science and Engineering, Vol. 4, 2012, 1146-1152.
9. Bakshi, A., B. Yogesh. Securing Cloud from DDOS Attacks Using Intrusion Detection System in Virtual Machine. – In: Second International Conference on Communication Software and Networks, 2010, 260-264.
10. Jung, J., B. Krishnamurthy, M. Rabinovich. Flash Crowds and Denial of Service Attacks: Characterization and Implications for CDNs and Web Sites. – In: WWW '02 Proceedings of the 11th International Conference on World Wide Web, ACM, 2002, 293-304.
11. Kandula, S., D. Katabi, M. Jacob, A. Berger. Botz-4-Sale: Surviving Organized DDoS Attacks that Mimic Flash Crowds. – In: NSDI ’05: 2nd Symposium on Networked Systems Design & Implementation, 2005, 287-300.
12. Wang, K., C-Y Huang, S-J Lin, Y-D Lin. A Fuzzy Pattern-Based Filtering Algorithm for Botnet Detection, Computer Networks. – The International Journal of Computer and Telecommunications Networking, Vol. **55**, 2011, 3275-3286.
13. Tripathi, A., A. Mishra. Cloud Computing Security Considerations, IT Division. DOEACC Society, Gorakhpur Centre, Gorakhpur, India, 2010, IEEE.
14. Sabahi, F. Cloud Computing Security Threats and Responses. – In: IEEE 3rd International Conference on Communication Software and Networks, 2011, 245-249.
15. Sabahi, F. Virtualization-Level Security in Cloud Computing. – In: IEEE 3rd International Conference on Communication Software and Networks, 2011, 250-254.
16. Moses. The Aussie Who Blitzed Visa, MasterCard and PayPal with the Low Orbit Ion Cannon, 9 December, 2010.
    http:// www.smh.com.au/ technology/ security/ the-aussie-who-blitzed-visa-mastercard- and -paypal-with-the-low-orbit-ion-cannon-20101209-18qr1.html
17. **http:// www.opnet.com/ news/ press_releases/ pr-2010/OPNET-Introduces-Cloud-Readiness-Service-pr.html**
18. **http://www.opnet.com./services/brochures/OPNETCloudReadiness.pdf**
19. Jha, R. K, U. D Dalal. On Demand Cloud Computing Performance Analysis with Low Cost for QoS Application. – In: International Conference on Multimedia, Signal Processing and Communication Technologies (IMPACT), 2011, 268-271.
20. Rakesh, K. J., U. D. Dalal. A Performance Comparison with Cost for QoS Application in On-Demand Cloud Computing. – IEEE Recent Advances in Intelligent Computational Systems (RAICS), 2011, 11-18.