A multilevel thrust filtration defending mechanism against DDoS attacks in cloud computing environment

N.Ch. Sriman Narayana Iyengar*

School of Computing Science and Engineering,
VIT University,
Vellore 632014, Tamil Nadu, India
Email: nchsniyengar48@gmail.com
*Corresponding author

Gopinath Ganapathy

Bharathidasan University Technology Park,
Trichy 620023, Tamil Nadu, India
Email: gganapathy@gmail.com

P.C. Mogan Kumar

School of Information Technology and Engineering,
VIT University,
Vellore 632014, Tamil Nadu, India
Email: mogankumarpc55555@gmail.com

Ajith Abraham

Machine Intelligence Research Labs (MIR Labs),
Scientific Network for Innovation and Research Excellence,
PO Box 2259, Auburn,
WA 98071-2259, USA
Email: ajith.abraham@ieee.org

Abstract: Cloud computing is a technology which not only gained advantages from ascendant technologies, but also suffered from its security breaches, of which availability is the most serious security issue. Distributed Denial of Service (DDoS) is a kind of resource-availability-related attack launched with the aim of subverting the Data Centre (DC) for resource unavailability to the legitimate clients. In this paper, we propose 'Multilevel Thrust Filtration (MTF) mechanism' as a solution, which authenticates the incoming requesters and detects the different types of DDoS attacks at different levels to detect the most intensive attack at an early stage to prevent the unnecessary traffic reaching the DC. A hybrid solution is proposed to detect four different kinds of attacks that have been taken into consideration. Profit analysis proved that the proposed mechanism is deployable at an attack-prone DC for resource protection, which would eventually result in beneficial service at slenderised revenue.

Keywords: cloud computing; availability; DDoS; distributed denial of service; flash crowd; botnet; spoof attack.

Reference to this paper should be made as follows: Iyengar, N.Ch.S.N., Ganapathy, G., Mogan Kumar, P.C and Abraham, A. (2014) 'A multilevel thrust filtration defending mechanism against DDoS attacks in cloud computing environment', *Int. J. Grid and Utility Computing*, Vol. 5, No. 4, pp.236–248.

Biographical notes: N.Ch. Sriman Narayana Iyengar (MSc, ME, PhD) is working as a Senior Professor at the School of Computing Science and Engineering at VIT University. He has 26 years of teaching and research experience. He has to his credit good number of publications. His research areas include cloud computing, networks (wired and wireless), intelligent computing and security.

Gopinath Ganapathy is a Professor and Head of Department of Computer Science and Engineering, Bharathidasan University, India. He received the Young Scientist Fellow Award for the year 1994 and eventually did research work at IIT Madras. He has published around 60 papers. He is a Member of IEEE, ACM, CSI and ISTE. He was a Consultant for ten years in international firms in the USA and the UK, including IBM, Lucent Technologies (Bell Labs) and Toyota. His research interests include modelling, patterns, NLP, web engineering and text mining.

P.C. Mogan Kumar is currently working as an Architecture for Cognitizent Technology Solutions in Chennai, India. He did his Postgraduation in Software Engineering from the School of Information Technology and Engineering, VIT University. His research interests include software engineering, distributed computing and security.

Ajith Abraham (Director, MIR Labs) received the MS from Nanyang Technological University (NTU), Singapore, and PhD (CSE) from Monash University, Melbourne, Australia. His research and development experience includes more than 23 years in the industry and academia. He works in a multidisciplinary environment involving machine (network) intelligence, cyber security, sensor networks, web intelligence, scheduling and data mining. He has given more than 70 conference plenary lectures/tutorials and invited seminars around the globe. He is an author/co-author of 900+ publications. He serves/has served the editorial board of over 50 international journals, guest-edited 40 special issues on various topics.

1 Introduction

Cloud computing, one of the most popular networking technologies, aims at cost-effective service provision scheme and combines the advantages of other ascendant technologies. Some of its unique characteristics like agility and multi-tenancy make this technology advantageous for adoption. Agility, a characteristic of cloud computing, allows a Cloud Service Provider (CSP) to support on-demand service provision scheme by reconfiguring the available resources. Multi-tenancy, another important characteristic of cloud computing, serves the same resource to multiple clients based on peak load, which eventually achieves optimal resource sharing and improved resource utilisation with effective cost. But these advantages throw in several security issues like data availability, data confidentiality and protection of the sensitive data maintained at the service provider's end.

The data availability issue deals with whether the subscribers' uploaded data can be downloaded at any time round the clock or whether it will be crashed by some anonymous external attackers or other competitor group subscribed for the same service to the same CSP. Regarding data confidentiality, the issue is whether subscribers' sensitive data reach the intended recipient. The issue regarding data protection is whether the encryption technique used by CSP is reliable enough to protect data residing in the Data Centre (DC). Though these issues have some considerable solutions, they still exist as a serious threat.

In cloud computing environment, data availability is maintained by deploying mirror servers which acts as a fault-tolerant mechanism. But we proposed a protocol with an intention to protect the DC against attackers and to maintain confidentiality with considerable data protection that resides in the DC. Firstly, in order to improve data availability, the DCs should be protected against any anonymous attackers' entry. So, the incoming requesters are to be authenticated. In order to uniquely identify and validate users, we use lightweight encryption technique.

Here, in our proposed work, we concentrate on data availability issues. There are several threats to data availability. Some of them are man in the middle attack, ping of death attack, ARP poisoning, smurf attack and Distributed Denial of Service (DDoS). Of all these, DDoS is the most serious threat, which is easy to launch and difficult to defend and prevent. This kind of attack is unusual. The sensitive data residing at the DC are corrupted and Service Level Agreement (SLA) is unsatisfied, which leads to cyber crime. Therefore, in order to protect the DC from such DDoS attacks, we have proposed a solution which uniquely and precisely validates each and every incoming resource requester at various levels for proving the requester's legitimacy. Only on passing the validations at all levels, will the requester be considered a legitimate requester.

The aim of our solution is to improve DC availability and allow the DCs to service only legitimate clients and to prevent other type of attackers' entry into the DC. This filtration achieves confidentiality, data theft prevention and DC resources protection, which ultimately result in improved throughput with negligible delay in traffic analysis.

The rest of the paper is organised as follows. Section 2 describes surviving techniques. Section 3 presents overview of the proposed architecture and methodology. Section 4 explains the working mechanism. Section 5 discusses the performance of the proposed mechanism. Section 6 deal with the advantages of the proposed approach and Section 7 provides the conclusions with an outline for our future work.

2 Surviving techniques

In Sabahi (2011), the Reliability, Availability and Security (RAS) issues of cloud computing are discussed. The CSP

should be able to provide the intended services and be able to manage the security from serious threats. The serious threat to reliability is data leakage. As the cloud data move from one tenant to several other tenants of the cloud, there is a serious risk of data leakage. Data leakage severity can be reduced by the use of Data Leakage Prevention (DLP). In order to maintain reliability, the DLP agents must be embedded into the cloud, but in a public cloud, it has less value of maintaining confidentiality, whereas in private cloud, the customers can take complete control over the cloud as it is designed for some group of people residing behind the firewall. In contrast, public cloud is used/shared by several users. Again, the serious threat to availability is DDoS. In order to avoid such powerful threats, cloud architecture should be designed in such a way to lessen the effect of such attacks, and the architecture should have the flexibility of instantaneous supply of resources to avoid service shutdown. The serious threat to security is unauthorised access of the cloud service. Non-repudiation of access control helps in legitimate access to applications, storage, operating systems and networks associated within the cloud.

To achieve flexible, scalable and efficient usage of resources, techniques such as partitioning, migration, workload analysis and allocation can be used for performance improvement in the virtual environment.

Hao and Han (2011) designed a new cloud architecture for improvement in data security. The architecture has several modules such as access control, buffer, security audit, classification write module and the final storage system. Firstly, the users are authorised by approving proper authentication method. The upload of data by the authorised users is observed by the request classification module to assess whether the data are ordinary data/sensitive data. Data classification is made by intelligent pre-fetching, data mining and feature extraction techniques. If the data are found to be ordinary data, they are stored directly to the storage system. If the data are found to be sensitive, they are stored in the buffer. The data are then analysed for content detection and for virus detection and then encrypted, which improves security and prevents the storage system from crashing. The metadata of ordinary data is secured in a metadata server1 as encrypted data. The metadata of sensitive data is secured in a metadata server2 as dual (double) encrypted data. The users can download after they are decrypted according to the type of data (ordinary/ sensitive).

Wang and Mu (2011) discussed security issues relating to network security, data security, lack of safety standards and information leakage repudiation. This paper also explains the characteristics of cloud computing such as dynamic scalability, virtualisation on a large scale, high availability, resource reuse and resource usage on demand.

Countermeasures to the above issues can be strengthening the anti-attack capability, information encryption, file encryption, protection of Application Programming Interface (API) keys, data backup and uniformity in standards and policies. These countermeasures help in retarding the effects of attacks, but the CSP should be reputed to ensure reliable gate keeping of user's information storage.

Du and Nakao (2010a) proposed DDoS filtering at network layer, so that the attack packets rate can be considerably reduced and the HTTP requests are allowed for further processing. Internet cloud contains intermediate node, which helps in identifying the threat without transferring it to the protected server.

In Chen et al. (2011), packet scoring and confidence-based filtering is used to identify threats, which acquire the nominal profile and also the attack profile, which has its own computation to predict whether a packet is legitimate or is an attack packet, based on the obtained packet score. Now, based on the score, the packets are allowed to access the server for further processing or filtered outside the network.

The following are the classic DDoS defence mechanisms, but they lack in defending cloud computing due to their limitations:

Black holing: It may be useful in networks where the attack packets and legitimate packets are differentiated, but it may not be useful if both attackers and legitimate packets enter into the cloud simultaneously because here in black holing the source address would not be informed regarding the failure in delivery of packet to the destined location.

Request rate throttling: The problem here is that there is no direct mapping between the number of requests per second and the number of open connections. But this scheme assumes uniform timing between requests. Even a slight randomness in the intervals leads to big changes in the number of open connections required to service this request rate. So the request rates could not be throttled at varying number of connections. Eliminating the incoming requests on exceeding the threshold, this has no mechanism of classifying the legitimate and attack requests.

Random dropping of request: Here the efficiency of processing the request is based on probabilistic chance, which may even deteriorate over different points of time. The efficiency cannot be guaranteed at high flooding of requests, which could consist of legitimate and spurious requests.

Sliding window protocol set-up: This involves sampling of incoming requests at any private network, so the comparison is made between sampled traffic and current traffic. Sampled traffic must be updated constantly. If efficient anomaly detection is not in place, this protocol will include spurious requests into current traffic detection.

Queuing the incoming request: This requires high queue bandwidth, which should be more than the legitimate request size. This scheme may disallow huge number of attack packets accessing the server, but this still queues up both legitimate and attack packets and creates delay in providing service to legitimate users.

In Joshi et al. (2012), the FDPM scheme was used to detect DDoS attacks. The scheme involves encoding procedure and reconstruction procedure. The packets sent by

the senders are marked with a code, which can be recovered by the reconstruction procedure to maintain authenticated transaction for each packet that reaches the server.

Du and Nakao (2010b) proposed over court gateways, which is a credit-based system where the well-behaving users will gain credit points and the ill-behaving users will lose their credit points. When the legitimate users exceed the threshold credit points, the users will be protected in a secure channel by path migration. When any users' characteristic leads to credit point exhaustion, the users will be blocked and they will not be able to access the server. This DDoS defence mechanism consists of one-hop path splicing, signalling mechanism, path migration, credit-counting system and path migration trigger.

Varalakshmi and Selvi (2010) proposed DDoS defence mechanism, which uses hop count filter, anomaly detectors, normal profile creation and attacker profile creation, and compares the incoming traffic to reduce false positive and false negative in order to improve the efficiency of attacker detection schemes using Kullback–Leibler divergence.

Intrusion Detection System (IDS) enhances the system by distributing the IDS nodes across the network. Host IDS collects audit data from the operating system. Network IDS collects data from the network packets. When any malicious intrusion is detected, the system generates reports and alerts. Fault-tolerant workflow scheduling makes use of failure probability information. Combining the heuristic information of tasks and replicating the tasks helps meet task deadline and saves resources.

Nesmachnow and Iturriaga (2013) proposed a solution for scheduling the independent tasks in heterogeneous computing. Raekow et al. (2013) proposed a licence management scheme in distributed environment, which authenticates the users to access the remote servers. This proposed solution is compatible to all the existing client-server architectures.

Raj Kumar and Selvakumar (2011) proposed Neural classifier which collects the incoming traffic and compared with the sample traffic. If the current traffic shows any deviation then the attack is detected, the attacks are classified as true positive, true negative, false positive and false negative. This classification improves the detection accuracy.

Janczewski et al. (2001) conveyed handling DDoS requires filtering the flooding attack and processing legitimate traffic. The idea proposed in the mechanism is to have a buffer and a filter greater than the size of the bandwidth, so that the even if the entire bandwidth is accommodated, the buffer could identify the incoming traffic and the DDoS flooding attack could easily be detected.

Chen et al. (2007) proposed an Anti-DoS (AID) scheme, which creates the overlay network for treating legitimate traffic, and attack traffic is another part of the overlay which requires special filtering treatment. So, the legitimate traffic is processed quickly, but the attack traffic is pre-processed at the overlay network. AID is a complete self-defence system, but the legitimate traffic is protected for server access.

According to Joshi et al. (2009), availability and privacy are serious issues for the dependants of cloud infrastructure. In order to satisfy customer requirement, the CSPs spend a huge amount of investment in redundancy of DCs. This is suitable to large DCs, but in the DC that maintains mid-class customers, enhancing such security leads to improper investment. The DDoS and worms and virus injection lead to loss of availability. Under such attacks, the users can immediately be migrated to the redundant DC, which is maintained as an additional DC, which is an image of the existing DC.

Al-Haidari and Salah (2011) proposed Economic Denial of Sustainability (EDoS), which is an attack that can be mitigated by employing a firewall that contains a white list and a blacklist. The authenticated users are given access and they are queued up at the white list, whereas the unauthenticated users are blacklisted and filtered at the firewall. The tables are updated periodically.

Table 1 Analysis of current surviving techniques

S. No	*Surviving techniques*	*Drawbacks/obsolete*
1	Chen et al. (2007) has proposed Anti-DoS (AID) mechanism.	Anti-DoS aims at detecting DDoS attacks but fails to classify the different types of DDoS attacks and also forgets to consider the spoofing kind of passive attacks.
2	Lo et al. (2010) proposed a cooperative intrusion detection system framework.	Intrusion detection system creates a network which IDS constantly alerts each other via cooperative agent. So, when any misbehaving attack scenario is found, it alerts. But this would prevent the attack from happening the next time and would protect the server from single point of failure attack.
3	Varalakshmi and Selvi (2010) proposed DDoS detection mechanism based on Kullback–Leibler divergence.	Kullback–Leibler divergence uses behaviour-based DDoS detection mechanism which helps in detecting DDoS attacks. This mechanism uses trust computation to rely on requester behaviour. This seems to behave with reduced detection efficacy with passive DDoS attackers.

Table 1 Analysis of current surviving techniques (continued)

S. No	*Surviving techniques*	*Drawbacks/obsolete*
4	Du and Nakao (2010a) proposed DDoS Detection Scheme based on Over Court Gateways.	Over court gateway mechanism is a credit-based accounting system which computes trust-based credit points to determine the requester behaviour. This mechanism detects the legitimate user and migrates to a separate channel. But this mechanism could not classify the impersonating passive DDoS attack.
5	Al-Haidari and Salah (2011) proposed EDoS detection mechanism.	EDoS add requester's IP as white-listed users and blacklisted users based on the behaviour. Blacklisted users would be filtered out at firewall, whereas white-listed users are allowed to access DC resources. But this scheme decides the behaviour instantly, which results in false positives.
6	Chen et al. (2011) proposed a packet filtering mechanism.	Packet filtering mechanism, namely confidence-based filtering which computes two different attack profiles (attack-period profile, non-attack-period profile). This mechanism lacks dynamism to detect the attack scenario when the network is extended.
7	Raj Kumar and Selvakumar (2011) proposed neural classifier.	Neural classifier uses behaviour-based DDoS detection scheme; it collects the sample data, processes the collected data, classifies the attack data and responds to legitimate users. This results in less false positives but suffers from computational overhead.
8	Jeyanthi and Iyengar (2012) proposed the Packet Resonance Strategy (PRS).	PRS has two levels of detection mechanism, which depends on one-time pass code, inter-arrival time and sealed sequence number. This mechanism attempts to detect DDoS attacks with efficiency consideration; it acquires intended communication channel for authenticated users.

All the above techniques lie under either router-based DDoS detection or host-based DDoS detection. The former technique helps in protecting network resource but takes too long a time for anomaly detection. The latter technique helps in protecting the resource but fails in protecting at very high rate of attack. We have analysed the advantages of both the schemes and the resource protection rate can be improved to make them effective even at the time of high DDoS attacks. This can be achieved by perfect synchronisation of DCs with our secured architecture-based implementation model. All the current surviving techniques either detect DDoS attack or create a secured architecture to protect the DC resource with certain a time lag. The proposed Multilevel Thrust Filtration (MTF) mechanism detects various kinds of attacks and authenticates them at different levels, and eventually makes the cloud environment free from attackers.

3 MTF: architecture overview

3.1 MTF: architecture

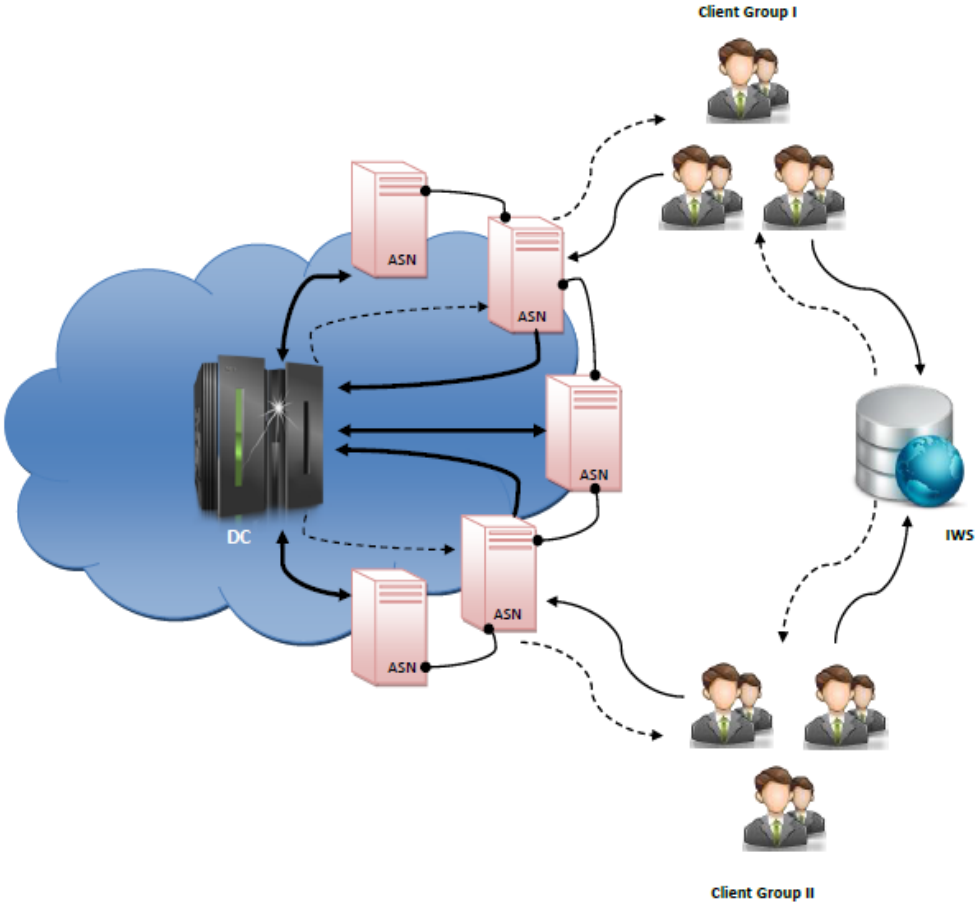
When any requester from any client group is interested in requesting a resource from the DC, they send the unique client ID to Intermediate Web Server (IWS), which acts as a look-up server. This IWS is maintained by CSP. So there is no need for any third party support for connecting the clients and DC. This IWS holds information about several DCs (shown as step 1 in Figure 1). When the requester requests the IWS, it finds out whether the incoming client ID is registered or not. If the ID is registered, the encrypted form of message is sent back to the client with the particular ASN's IP address, which is ciphered with clients' password (shown as step 2 in Figure 1). Now, the ciphered message recipient will be able to decrypt the ASN IP address only if the recipient is the intended client, so that the client could communicate further. Any other requester is considered a fake requester. On consecutive failures for a certain number of times, the requester is filtered at the firewall.

On successful decryption, the client forwards the message sent by the IWS to the ASN (shown as step 3 in Figure 1). Now the ASN validates the digital signature of the IWS by decrypting it. On successful validation, a certificate is generated which contains the session key and timestamp. This is valid only for the particular concerned client. This generated certificate is stored at the ASN and also sent to the client (shown as step 4 in Figure 1). When the client decrypts the ASN's response, the client gets the session key and communicates with the DC via the ASN (shown as steps 5 and 6 in Figure 1).

Here, the ASN acts as an exclusive protection layer which the requester is unaware of. So the requests cannot bypass the ASN and reach the DC. This ASN protection chain helps in deploying the detection mechanism at the cloud boundary to detect the abnormal traffic condition earlier, which is much cost-effective.

The proposed MTF has been categorised into four modules for detecting and preventing attackers' entry into the cloud environment: traffic analysis, abnormality detection, abnormality classification and attack prevention.

Figure 1 Architecture of MTF scheme (see online version for colours)



Notes: Solid lines: requests; Dotted lines: responses; Thick solid lines: requests and responses between Authorised Scrutinising Node (ASN) and DC; Dotted arrow: connections between ASN.

3.1.1 Traffic analysis

Whenever requesters request for the DC resource, they are bypassing IWS and ASN, so at every point of time each ASN captures the request rate and request size of each incoming requester. Since the DDoS attacker and legitimate requester send the same message request, they have the same message structure, but they vary in their traffic patterns. This differentiation is enough to measure the traffic condition and identify the attack group. DDoS attackers aim to shut down the DC by exploiting the DC's resource, which prevents the legitimate requester from accessing the server resource, and the DC resource becomes unavailable to the legitimate users. Since attackers imitate the legitimate users in sending the request, they cannot be differentiated in the message patterns, but they can be differentiated in their traffic patterns.

3.1.2 Abnormality detection

On successful validation of digital signature, the request rate and request size are logged for each particular requester. If the incoming requester follows the nominal traffic profile, he/she is served. In case the requester misbehaves, the requester with an abnormal traffic pattern is considered an attacker. A requester is also considered an attacker if the request size of any individual requester exceeds the maximum legitimate request size.

3.1.3 Abnormality classification

When any requesters' request rate or request rate varies, abnormality is identified. But the cause of the abnormality has to be found in order to predict whether the abnormal traffic is due to flash crowd or DDoS attackers.

Flash crowd is an event that occurs when a large number of legitimate users try to access the server resources simultaneously, but this traffic congestion is for a short period of time and the request rate for each legitimate user does not exceed the maximum request rate. DDoS attack, on the other hand, is a kind of attack that attempts to completely subvert the server resources where the request rate and request size vary from the nominal profile.

3.1.4 Attack prevention

By examining the request rate and request size at the time of traffic abnormality, it is easier to differentiate between the attackers and the legitimates. The attack behaviour of the requester is informed to the IWS by the ASN and considered as an attacker. So the IWS restricts further requests from that particular attacker for that particular

session. The attacker requests are now filtered at the firewall. Thus, the attack behaviour can be prevented.

Our architecture aims at absolute authentication of the requesters, which is based on cipher without creating overhead and uniquely identifying each requester at the initial stage. Since the DC resources are essential, when it is locked by an anonymous attacker, the legitimate user could suffer resource unavailability. Since cloud computing environment has a vast resource for servicing the legitimate, the client group will be billed for the DDoS attackers' activities.

The architecture aims at capturing the four different kinds of traffic congestions at different levels that could affect the performance of the service provider. The four kinds of traffic congestions are, firstly, spoof attack, secondly, DDoS attack, thirdly, flash crowd event and, finally, aggressive legitimates. The working mechanism and remedial measure for different kinds of attacks will be described in detail in Section 4.

4 MTF: working mechanism

The detailed working mechanism of MTF mechanism is shown as a working procedure in Protocol 1.

4.1 Protocol of MTF mechanism

Protocol 1 Protocol of our proposed approach

At the time of Authentication Phase

- Client ID Recognition
  - **Client→IWS:** ID_{Client}, $ID_{RequestType}$
  - **IWS→Client:** $E(K_{IWS\_ASN}[ID_{Client}, Password_{Client}, ID_{ClientMAC}, ID_{RequestType}]) \,||\, E(K_{IWS\_CLIENT}[ID_{ASN}])$
  - **At Client:** $D(K_{IWS\_CLIENT}\ E(K_{IWS\_CLIENT}[ID_{ASN}]))$
- Digital signature Validation
  - **Client→ASN:** Forwards $E(K_{IWS\_ASN}[ID_{Client}, Password_{Client}, ID_{ClientMAC}, ID_{RequestType}])$
  - **At ASN:** $D(K_{IWS\_ASN}, E(K_{IWS\_ASN}[ID_{Client}, Password_{Client}, ID_{ClientMAC}, ID_{RequestType}]))$
  - **ASN→Client:** Certificate is generated, stored & forwarded to client and service processed by ASN→ DC $E(Password_{Client}, [K_{SESSION}, CEP, ID_{RequestType}, TS])$
  - **At Client:** Acquire Certificate $D(Password_{Client}, E(Password_{Client}, [K_{SESSION}, CEP, ID_{RequestType}, TS]))$
- Request Acquisition
  - **ASN→DC:** $E(K_{ASN\_DC}[ID_{Client}, K_{SESSION}, ID_{RequestType}, Password_{Client}])$
  - **DC→ASN:** $D(K_{ASN\_DC}, E(K_{ASN\_DC}[ID_{Client}, K_{SESSION}, ID_{RequestType}, Password_{Client}])) \,||\, E(K_{ASN\_DC}[ID_{Client}, K_{SESSION}, Status_{APPROVAL}, Password_{Client}])$
  - **At ASN:** $D(K_{ASN\_DC}, E(K_{ASN\_DC}[ID_{Client}, K_{SESSION}, Status_{APPROVAL}, Password_{Client}]))$

After Authentication Phase for each incoming request

- Behaviour Monitoring
  - **Client→ASN:** $ID_{RequestType}$, $K_{SESSION}$
  - **At ASN:** Monitor $Rate_{Request}$, $Size_{Request}$, $Type_{Request}$

```
IF (! Abnormal behaviour) {
   Service Provision
}
ELSE{
   Transient Ejection
}
```

- Service Provision
  - **ASN→DC:** $ID_{RequestType}$, ID_{Client}, $K_{SESSION}$
  - **DC→ASN:** Requested service served (Email, File Download) $ID_{RequestType}$, $K_{SESSION}$, TS
- Transient Ejection
  - **ASN→Firewall:** ID_{Client}
  - **At Firewall:** Filters ID_{Client} for $K_{SESSION}$
  - **ASN→IWS:** $E(K_{IWS\_ASN}[ID_{Client}])$
  - **At IWS:** Blocks ID_{Client} $D(K_{IWS\_ASN}, E(K_{IWS\_ASN}[ID_{Client}]))$

Legends: ID_{Client} → client's ID; $ID_{RequestType}$ → type of incoming request (new/ registered client); K_{IWS_ASN} → secret key between IWS and ASN**;** ID_{ASN} → ASN's IP address; $Password_{Client}$ → client's password; $ID_{ClientMAC}$ → vlient's MAC address; $K_{SESSION}$ → session key; CEP → certificate expiration time; K_{ASN_DC} → secret key between ASN and DC; $Status_{APPROVAL}$ → request status approval at DC; $Rate_{Request}$ → request rate of incoming client; $Size_{Request}$ → request size of incoming client; $Type_{Request}$ → application request type of incoming client; TS → timestamp; E → encrypt; D → decrypt.

4.1.1 Level I: link pre-fetch

Whenever requesters try to request any resource at the DC, they are authenticated. In the authentication phase, the requesters are uniquely identified and validated, and their request types are registered.

Initially, the clients must be authenticated at the IWS. The IWS acts as a look-up server which identifies the registered requesters by their IDs. Then, it immediately checks the ASN with least load and shoots out the message to the client. This message contains two pieces of information. One is for the client and the other is for the ASN. The information for the client is the ASN IP address, which is encrypted with the secret key shared between the client and the IWS. This achieves privacy and confidentiality. The information for the ASN is the requester ID, requester password, requester MAC address and the request type, which is encrypted with secret key shared between the IWS and the ASN.

Now, when the requester reaches the ASN successfully, the requester is uniquely recognised but still needs to be authenticated.

4.1.2 Level II: requester scrutinising

The next step in authentication is at the ASN. Now, the ASN authenticates the IWS indirectly via the message forwarded by the requester, which is shown as a part of the digital signature in the above protocol. The ASN decrypts the message and obtains the requester's ID, password, MAC address and request type. This helps in achieving privacy and confidentiality between the IWS and the ASN.

This also assures that the requester is recognised at the IWS and bypasses the IWS. On validating the digital signature, a certificate is generated for the requester, which is both stored at the ASN and sent to the client by encrypting it with the client's password. So only the intended client can view the certificate and its contents.

The contents of certificate are:

- Client ID: uniquely identifies the registered clients,
- Session key: uniquely identifies the active clients connected at the DC,
- Certificate expiration period: time until which the current connected clients' are valid,
- Timestamp: for authenticating the client's connection based on the synchronised time.

4.1.3 Level III: traffic data logging

Once the certificate is generated for the requester, the requester is considered to be authenticated and becomes a legitimate client. So, from now on, the traffic data are monitored for each incoming client. If the request rate or request type is inappropriate, the concern client is notified and warned for a certain number of times. If the misbehaviour continues, then the client is outwitted until the session expires.

Pseudocode 1 Traffic data logging

```
For each client
  For each Incoming request
    If (Request Type && Request Size are appropriate)
      If (Request Rate is appropriate) {
          Allowed In for further Processing.
        }
        Else {
          If (Inappropriate Request Rate found < N) {
            Overload attack prone attempt is identified and the client is warned.
          }
          Else {
            Overload attack attempt is confirmed and client filtered from accessing DC resources for that particular session
          }
      }
```

Here N = maximum number of acceptable request rate violation. N depends on environment conditions like attack-prone zone and network traffic. N is inversely proportional to attack-prone zone. If the DC located at a highly attack-prone zone or frequent attack-prone zone, then it is advisable to set N near to 1. Otherwise, the maximum range can be 5.

The reason for allowing the request rate violation is because the clients are examined thoroughly, authenticated and approved as legitimates. The slight variation in request rate should not outwit the client immediately; rather the client is warned and monitored for any further violations of the same nature. If it continues for N times, then the concern client can be outwitted. So this clarifies each client's traffic behaviour, which can be analysed in Pseudocode 1.

4.1.4 Level IV: access right approval

The access right approval is of two kinds: (1) service provision and (2) transient ejection.

Access right approval is required (1) immediately after the initial authentication and (2) during traffic behaviour monitoring.

Service provision at initial authentication: when the client decrypts the certificate and obtains the session key, the service is provisioned.

Transient ejection at initial authentication: when the client could not obtain session key within certificate expiration period, the client is recognised as an anonymous attacker and filtered at the firewall.

Service provision at traffic behaviour monitoring: at this stage, the clients are authenticated as legitimate; if the traffic data log at the ASN matches the legitimate request traffic behaviour, the incoming requests are forwarded to the DC and the service provisioned for the client.

Transient ejection at traffic behaviour monitoring: initial authentication identifies the legitimate behaviour, but that is not enough to protect the DC resource, because the legitimate may also create an overload attack scenario. So it has to be identified precisely and outwitted. For doing so, continuous traffic behaviour of each client is monitored and if any abnormality is found, the concern client is warned. After a certain number of legitimate behaviour violations, the client is outwitted permanently until the session expires.

4.2 Issue resolving solutions

MTF is a mechanism which learns the traffic behaviour of each client and identifies the abnormal traffic condition and protects the DC resources from exhaustion. Our MTF helps in resolving the following four different types of attacks: botnets, DDoS, flash crowd and spoofing attack.

4.2.1 Type 1: botnets

Botnet (Robot Network) is a group of computers that injects malicious requests towards the DC and makes the DC resources unavailable to legitimate users.

Since the botnets attempt to keep on overloading at the DC, they cause the highest degree of overload threat. When they are detected at the earliest, the unnecessary traffic is reduced which paves the way for the legitimates. So whenever the botnets try to attack the DC, they simply overload the junk requests towards the DC. But they reach the IWS before they reach the DC. So at the IWS, a ciphered response is sent to the requester (presently botnets). But the IWS cannot receive any response because botnets are programmed to inject malicious requests, which blocks any responses to the IWS, they keep on overloading with the same MAC address, which is also another clue to detect them.

4.2.2 Type 2: DDoS

DDoS attacks are initiated and continued by some hundreds of attackers which start populating the unwanted traffic packets with enormous size in order to acquire the memory resources and completely deplete them. By the way, this traffic stops the legitimate requests from reaching the data centre and depletes the bandwidth of the data centre. This, at some point of time, leads to unresponsiveness to its legitimate requests.

DDoS is another overload threat of the highest degree, which involves several human attackers to subvert the DC. This kind of attackers aim at degrading the DC performance as quickly as possible, so they will attempt to overload with the legitimate request type, but their behaviour (traffic request rate) will be different. Because their aim is to exhaust server resources, the request rate will definitely deviate from that of the legitimates, which is a clue to detect these kinds of overload threats.

4.2.3 Type 3: flash crowd

Flash crowd is an overload condition caused by simultaneous incoming of large number of legitimates over a short period of time.

Flash crowd is an overload threat but does not create any harm to the DC. This kind of overload occurs when a huge number of legitimate individuals rush towards the DC simultaneously. But this will not affect the DC resource and will not continue for a prolonged period of time. After authentication at the IWS and the ASN, the request rate for each individual legitimate will be negligible. The difference between DDoS and flash crowd is the former involves a large number of individuals who violate legitimate request rate and causes overload, whereas latter involves a large number of legitimate individuals who do not violate the request rate.

4.2.4 Type 4: spoofing attack

This is a kind of overload condition, which is an attempt by an attacker by impersonating the legitimate user and affecting the DC resource by illegitimate access.

This kind of attack has the intent of data stealing or data corruption at the DC. This kind of attackers will behave like the legitimates, but this attacker can be easily detected at the IWS. The ciphered response from the IWS is based on the password of the incoming requester. So, in case of a spoof attack, the response from the IWS cannot be revealed as it was ciphered.

5 Experimentation and performance evaluation

5.1 Experimental set-up

Jeyanthi and Iyengar (2012) used OPNET as simulator to test the cloud computing environment (http://www.opnet.com/news/press_releases/pr-2010/OPNET-Introduces-Cloud-Readiness-Service-pr.html, accessed on 17 July 2013). Jeyanthi et al. (2013) experimented DDoS in cloud computing. Jha and Dalal (2011) has experimented for pricing the cost incurred at Quality of Service (QoS) for on-demand cloud computing. We tested our proposed mechanism as simulation experiment in OPNET Modeler (http://www.opnet.com./services/brochures/OPNET_Cloud Readiness.pdf, accessed on 17 July 2013). The experiments are performed in a campus network where DC requesters are grouped in three subnets and each subnet has got 100

workstations, 100 attackers and 200 legitimate clients requesting for application-specific requests at each subnet. This way we created the attacker and legitimate profile and other devices, which would be needed to test our algorithm as an experiment. The traffic represents internet and the group of spoof attackers is activated at varying time intervals. The attack profile is replicated to increase the attack strength to engage the DC resources like bandwidth, CPU and memory. On the whole, our experiment has 600 clients and 300 attackers. The experiment is carried out with three different scenarios, namely the network with no attackers, networks with attackers and no detection mechanism in place, and finally the network with attackers and the proposed MTF mechanism in place.

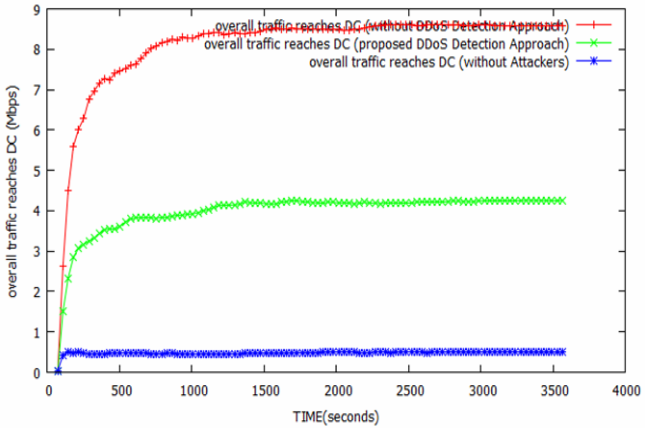
5.2 Performance evaluation

5.2.1 Traffic rate at DC

Traffic rate is the average number of bits forwarded per second to the email application, File Transfer Protocol (FTP) application and HyperText Transfer Protocol (HTTP) application towards the DC. Flooding traffic rate generated by distributed attackers is identified towards the victim data centre.

Figure 2 shows the overall traffic approaching towards the DC. Here, the traffic rises suddenly, which shows the DDoS attack is launched. When launched, all the attackers flood the packets towards the DC, which is shown in Figure 2. These traffic data at different scenarios are tracked, captured and plotted to prove that traffic approaching towards the DC is reduced considerably from highly busy network with a traffic rate of above 8 Mbps to around 4 Mbps being achieved. So preventing the traffic attacks towards the DC protects the DC resources, which eventually proves the effectiveness of our proposed solution.

Figure 2 Overall traffic approaching DC (see online version for colours)

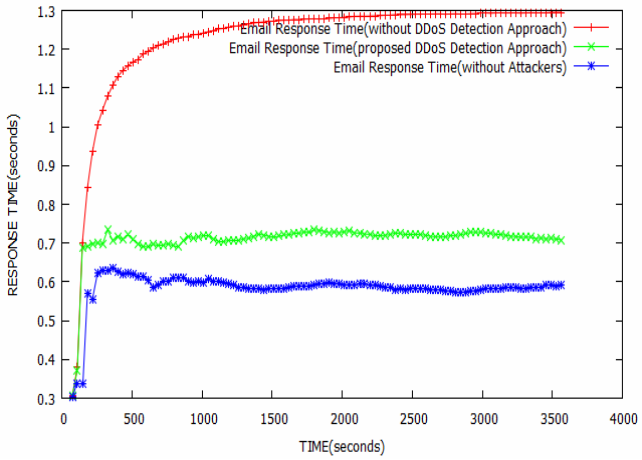


5.2.2 Email response time

Email response time is the statistic measured as the time elapsed between the sending of requests for emails and receiving emails from the email server in the network. This time includes signalling delays for the connection set-up.

Figure 3 shows the response time of the email application with MTF mechanism and without MTF mechanism. The steep increases in response time shows that the DDoS attackers create network overhead and congest the DC, which results in prevention of the legitimates to be serviced. The service provisioned by the DC at the time of DDoS is improved twice, which from 1.3 seconds to 0.7 seconds. The email data transfer is about 100 Kb–500 KB.

Figure 3 Email response time (see online version for colours)



5.2.3 FTP response time

FTP response time is the statistic that measures the time elapsed between sending a request and receiving the response packet. It is the measure of the time from when a client application sends a request to the server to the time it receives a response packet. Every response packet sent from a server to an FTP application is included in this statistic.

Figure 4 shows the response time of FTP application with MTF mechanism and without MTF mechanism. Here, the proposed MTF shows its efficiency in serving better at the time of DDoS. Because of FTP application, the file download requests are about 500 Kb–5 Mb. But in the proposed MTF mechanism, even at the time DDoS, the victim DC can serve the legitimate user and can prevent the attacker requests, which are usually at a higher rate.

Figure 4 FTP response time (see online version for colours)

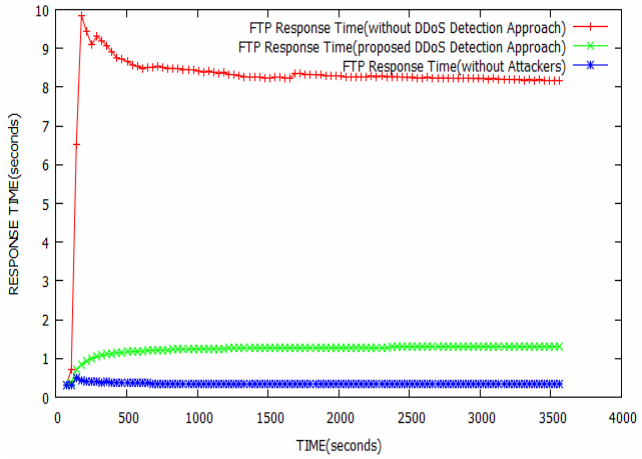


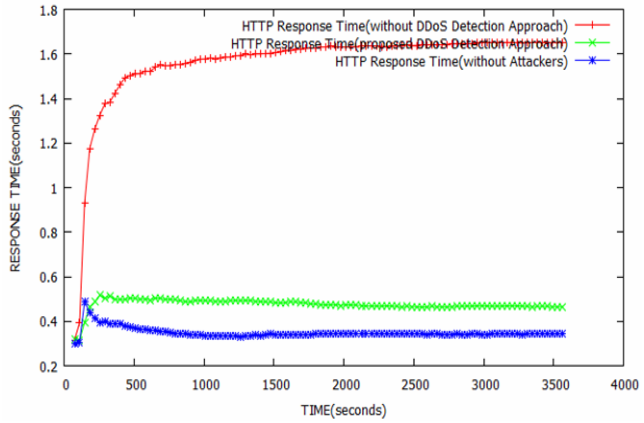
Figure 4 shows that the MTF mechanism proves to be seven times more efficient in responding to legitimates even at the time of DDoS.

5.2.4 HTTP response time

HTTP response time is a statistic that specifies time required to retrieve the entire page with all the contained inline objects. This statistic also includes the response time for each inline object from the HTML page.

Figure 5 shows the response time of HTTP application with MTF mechanism and without MTF mechanism. HTTP application requests ranges from 10–200 Kb in size. In Figure 5, the deployment of MTF mechanism proves that it is three times better at the time of DDoS and a noticeable point is that the application traffic almost behaves as it was without attackers.

Figure 5 HTTP response time (see online version for colours)



5.2.5 Bandwidth resource utilisation

Bandwidth resource utilisation is the statistic that represents the average number of bits received or transmitted successfully by the receiver or transmitter channel per unit time, in bits per second. As the traffic includes both legitimate and attack patterns, we consider only the legitimate data traffic that reaches the DC and record at each transaction.

Figure 6 shows the bandwidth utilised at the DC with MTF mechanism and without MTF mechanism. Since at the time of DDoS the legitimates fail to access resources at the DC, it leads to retransmissions and creates increased traffic which also affects the response time that has been shown in Figures 3–5. Once when the attackers are blocked and prevented from entering into cloud environment, then the bandwidth resource is protected which can be seen in Figure 6.

5.2.6 User connections cancellation

The number of connections cancelled is the statistic that measures the total number of legitimate connections aborted by huge traffics, increasing the number of connections at each time a TCP connection is aborted at this node.

Figure 6 Bandwidth resource utilisation (see online version for colours)

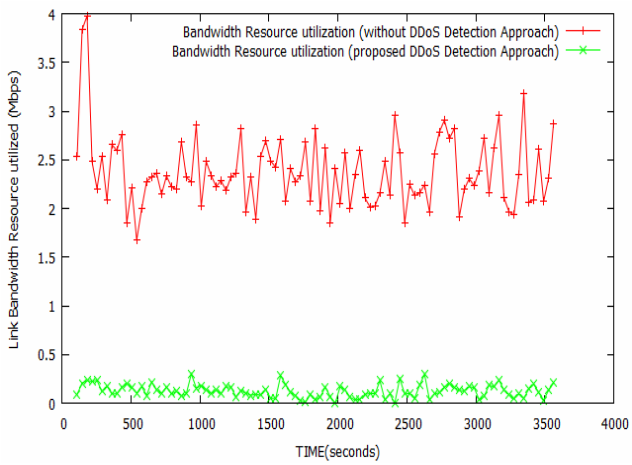
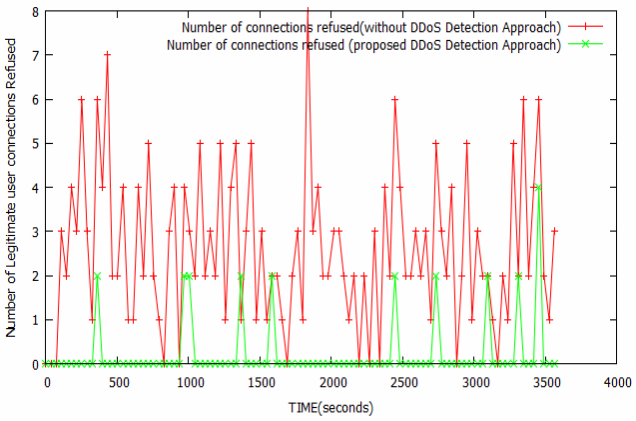


Figure 7 shows the user connections cancelled at the DC with MTF mechanism and without MTF mechanism. The number of conflicts and legitimate use connections cancelled can be seen in Figure 7. It also proves that the deployment of MTF mechanism almost protects the user connections and their session, which proportionately improves to respond the legitimate at the earliest.

Figure 7 User connection cancelled (see online version for colours)

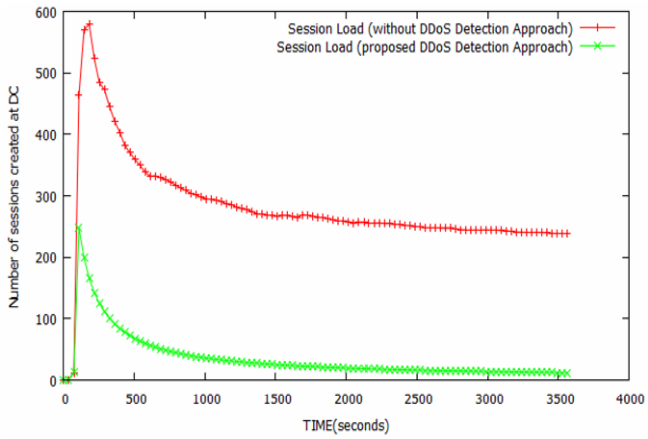


5.2.7 Sessions creation at DC

Session task load represents the current number of application sessions on the DC. This statistic is intended to provide a picture of how much loaded the server is with application sessions.

Figure 8 shows the session load at the DC with MTF and without MTF mechanism. The number of sessions without MTF shows the session load that was created and maintained for both legitimate and DDoS attackers. Since attackers also maintain sessions, the DC struggles to serve the legitimates with the attacker loads, which could not be detected. When the MTF mechanism is deployed, the attackers are outwitted initially which is shown as an initial step and later, only the legitimate sessions are maintained.

Figure 8 Sessions created at DC (see online version for colours)



6 Advantages of proposed solution

6.1 Profit analysis

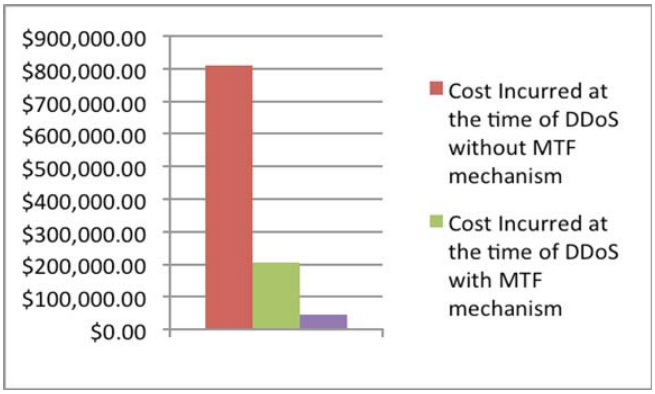
The cost is computed based on the data transmission and memory resident operations at each DC, based on an average sample that is a combination of attack traffic and legitimate traffic.

Let N = time in hours; C_{BW} = bandwidth cost; C_{MEM} = RAM cost of each physical equipment; C_{VM} = VM cost of each physical equipment and C_{DS} = data stored within a DC:

$$\text{Total cost incurred at DC} = \sum_{i=1}^{N} \{C_{BW} + C_{MEM} + C_{VM} + C_{DS}\}$$

Figure 9 shows the huge cost incurred at the victim DC. When the proposed MTF is in place, the cost incurred at the DC improved *revenue*, which results in resource protection and is used only for legitimates that in turn results in resource availability. The costs used are \$ 0.1/Gb for any data transmission at the DC and \$ 0.05/second for any memory resident operations at the DC. The extreme difference in profit is due to detection of attacker at their initiation and preventing their subsequent entry towards the DC. This paves the way to improve availability with an acceptable response time shown in Figures 3–5. In addition to the improved detection efficacy, other benefits have been observed that would improve the choice of deployment.

Figure 9 Profit analysis (see online version for colours)



6.2 Advantages of MTF mechanism

Hierarchical detection: the proposed mechanism detects different overload conditions at different levels. Initially at the IWS, botnets and spoofing attacks are detected. Later at the ASN authentication phase, spoofing attacks are detected. By deploying the traffic log, DDoS and flash crowd is detected and distinguished precisely.

Reduced traffic: the detection at the earlier stage helps the attacker to be outwitted earlier, which considerably reduces the traffic at the DC. This proportionately allows legitimate traffic inwards and makes the detection mechanism advantageous when deployed by CSP.

Hybrid approach: the proposed MTF approach combines both the host-end deployment and router-end deployment solutions, because MTF needs the ASN deployed at the host end and also authenticates the incoming requester at router level through IWS.

Assured security requirements: the use of the cipher achieves privacy, confidentiality and precisely authenticates the incoming requester, which satisfies the security requirements. This does not create overhead to the detection mechanism because the cipher is used only for the initial authentication purpose.

7 Conclusion and future work

DDoS attacks are very common attacks to exhaust the resource of the DC. This attack is easier to launch and difficult to detect. So it is necessary to deploy the detection mechanism which identifies each requester and their incoming traffic rate to detect whether the incoming requester is an attacker or a legitimate user. The proposed MTF mechanism makes use of the authentication protocol that involves cipher, which uniquely identifies the requester and lets him in. One of the advantages is that cipher is used at the initial stage. Thereafter, the cipher is not involved, so there is no chance of overhead in the detection scheme. Moreover, it also detects and filters four different kinds of overload conditions such as botnet, DDoS, flash crowd and spoof attacks.

This proposed solution is neither a completely host-based solution nor a router-based solution. Each has its own advantages. We combined the advantages of both and made it better in detecting attackers. Our future work is to detect the DNS amplification attacks using the proposed mechanism.

References

Al-Haidari, M.H. S.F. and Salah, K. (2011) 'EDoS-shield: a two-steps mitigation technique against EDoS attacks in cloud computing', *Proceedings of the 4th IEEE International Conference on Utility and Cloud Computing*, 5–8 December, Victoria, NSW, pp.49–56.

Chen, Q., Lin, W., Dou, W. and Yu, S. (2011) 'CBF: a packet filtering method for DDoS attack defense in cloud environment', *Proceedings of the Ninth IEEE International Conference on Dependable, Autonomic and Secure Computing*, 12–14 December, Sydney, NSW, pp.427–434.

Chen, S., Ling, Y., Chow, R. and Xia, Y. (2007) 'AID: a global anti-DoS service', *Computer Networks*, 2007, pp.4252–4269.

Du, P. and Nakao, A. (2010a) 'DDoS defense as a network service', *Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS 2010)*, 19–23 April, Osaka, Japan, pp.894–897.

Du, P. and Nakao, A. (2010b) 'OverCourt: DDoS mitigation through credit-based traffic segregation and path migration', *Computer Communications*, Vol. 33, No. 18, pp.2164–2175.

Hao, L. and Han, D. (2011) 'The study and design on secure-cloud storage system', *Proceedings of the International Conference on Electrical and Control Engineering (ICECE)*, 16–18 September, Yichang, China, pp.5126–5129.

Janczewski, L.J., Reamer, D. and Brendel, J. (2001) 'Handling distributed denial-of-service attacks', *Information Security Technical Report*, Vol. 6, pp.37–44.

Jeyanthi, N. and Iyengar, N.C.S.N. (2012) 'Packet resonance strategy: a spoof attack detection and prevention mechanism in cloud computing environment', *International Journal of Communication Networks and Information Security*, Vol. 4, No. 3, pp.163–173.

Jeyanthi, N., Iyengar, N.C.S.N., Mogan Kumar, P.C. and Kannammal, A. (2013) 'An enhanced entropy approach to detect and prevent DDoS in cloud environment', *International Journal of Communication Networks and Information Security*, Vol. 5, No. 2, pp.110–119.

Jha, R.K. and Dalal, U.D. (2011) 'A performance comparison with cost for QoS application in on-demand cloud computing', *Proceedings of the IEEE Recent Advances in Intelligent Computational Systems (RAICS),* 22–24 September, Trivandrum, pp.11–18.

Joshi, B., Vijayan, A.S. and Joshi, B.K. (2012) 'Securing cloud computing environment against DDoS attacks', *Proceedings of the International Conference on Computer Communication and Informatics (ICCCI)*, 10–12 January, Coimbatore, India, pp.1–5.

Joshi, K.R., Guy, B., Jahanian, F., van Moorsel, A.P.A. and Weinman, J. (2009) "Dependability in the cloud: Challenges and opportunities", *Proceedings of the 2009 IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2009*, 29 June–2 July, Estoril, Lisbon, Portugal, pp.103–104.

Lo, C-C., Huang, C-C. and Ku, J. (2010) 'A cooperative intrusion detection system framework for cloud computing networks', *Proceedings of the 39th International Conference on Parallel Processing Workshops (ICPPW)*, 13–16 September, San Diego, CA, pp.280–284.

Nesmachnow, S. and Iturriaga, S. (2013) 'Multiobjective grid scheduling using a domain decomposition based parallel micro evolutionary algorithm', *International Journal of Grid and Utility Computing*, Vol. 4, No. 1, pp.70–84.

Raekow, Y., Simmendinger, C., Jenz, D. and Grabowski, P. (2013) 'On-demand software licence provisioning in grid and cloud computing', *International Journal of Grid and Utility Computing*, Vol. 4, No. 1, pp.10–20.

Raj Kumar, P.A. and Selvakumar, S. (2011) 'Distributed denial of service attack detection using an ensemble of neural classifier', *Computer Communications*, Vol. 34, No. 11, pp.1328–1341.

Sabahi, F. (2011) 'Cloud computing security threats and responses', *Proceedings of the IEEE 3rd International Conference on Communication Software and Networks (ICCSN)*, 27–29 May, Xi'an, China, pp.245–249.

Varalakshmi, P. and Selvi, S.T. (2013) 'Thwarting DDoS attacks in grid using information divergence', *Future Generation Computer Systems*, Vol. 29, No. 1, pp.429–441.

Wang, J. and Mu, S. (2011) 'Security issues and countermeasures in cloud computing', *Proceedings of the IEEE International Conference on Grey Systems and Intelligent Services (GSIS)*, 15–18 September, Nanjing, China, pp.843–846.