Bi-level user authentication for enriching legitimates and eradicating duplicates in cloud infrastructure

R. Thandeeswaran*

School of Information Technology and Engineering,
VIT University,
Vellore, Tamilnadu, India
Email: rthandeeswaran@vit.ac.in
*Corresponding author

M.A. Saleem Durai

School of Computer Science and Engineering,
VIT University,
Vellore, Tamilnadu, India
Email: masaleemdurai@vit.ac.in

Abstract: Ease of usage of cloud computing leads to an exponential growth in all sectors. Exponential growth always attracts duplicates to consume and deplete resources. Cloud is not exempted from invaders and overwhelming the resource utilisation thereby availability become a threat. Availability issue arises due to multiple requests towards the same victim, a DDoS attack. Hence, the major concern in the cloud is to rightly identify legitimates, and providing the required services all time go by avoiding DDoS attacks. Multiple techniques are available to identify and authenticate the users. This paper not only just tries to authenticate the users but also works on eliminating the invaders in two fold. In the first phase, the user ID is scrambled in four different steps. In the second phase, the users are authenticated depending on the credits. Based on the traffic flow (in the case of network level attack) and on the interval between consequent service requests (in the case of service level attack), users are authenticated upon which services are provisioned accordingly. The simulation results presented here exhibits the strength of the proposed method in detection and prevention of DDoS attack in cloud computing environment.

Keywords: DDoS attack; SSID; authentication; credits; cloud environment; legitimate; attackers.

Reference to this paper should be made as follows: Thandeeswaran, R. and Saleem Durai, M.A. (2020) 'Bi-level user authentication for enriching legitimates and eradicating duplicates in cloud infrastructure', *Int. J. Computer Aided Engineering and Technology*, Vol. 12, No. 1, pp.95–112.

Biographical notes: R. Thandeeswaran is a Research Scholar cum Faculty in the School of Information Technology and Engineering, VIT University, Vellore, India. He received his Bachelors in Computer Engineering from Madurai Kamaraj University and Masters in Computer Science and Engineering from VIT University, Tamil Nadu, India. He published papers in national and international journals and conferences. His areas of interest include network security, cloud computing and software project management. He is a life member in Computer Society of India.

M.A. Saleem Durai received his PhD from VIT University, Vellore, Tamilnadu, India in 2011. He is an Associate Professor in the School of Computing Science and Engineering at VIT University, Vellore, Tamilnadu, India. He has authored many international and national journal papers to his credit. His research interests include data mining, fuzzy logic, cloud computing and rough sets. He is associated with many professional bodies CSI and IEEE.

1 Introduction

Distributed denial of service (DDoS) attacks is the most hazardous of all, in a cloud environment. This kind of attack stifles the cloud service provider (CSP) in a way that the data centre resources will get exhausted and it fails to serve the service request from legitimate users. The attackers usually execute this by compromising insecure systems distributed across the network and install malware applications in those systems which are capable of sending multiple requests to the CSP simultaneously. The DDoS attacks can occur at two different levels, network level and at service level.

- In network level DDoS, the attackers will try to send some invalid requests with the aim of flooding the CSP; e.g., requests for a half-open connection.
- In service level DDoS, the attacker will be sending requests that seem to be legitimate. Their content will be similar to a request made by a legitimate user. Only their intention is malicious. But, CSP would not be able to discriminate this kind of requests for services.

Thus, it is high time now to devise some techniques to eradicate these two kinds of DDoS attacks and make the cloud 100% secure from DDoS.

In 2012, the whistle-blowing website WikiLeaks had been flooded with ten gigabits per second of traffic, making it slow and unresponsive. In 2009, the Bitbucket, a web-based code hosting company experienced more than 19 hours of downtime due to a DDoS attack. They were using Amazon EC2 service. They became victims of the same twice (Jeyanthi et al., 2014). At first, the attacker used UDP packets for flooding and next day after the blocking the massive UDP packets the attacker launched the DDoS attack using TCP SYN requests. Even though Amazon recovered, the Bitbucket decided to move to some other CSP who poured oil on the burning fire. The DDoS attack was using spoofed packets and hence the originator couldn't be yet traced.

Scalability, which is one of the most important features of the cloud, is also a vulnerability that makes a bed for DDoS attacks. As the number of requests increases, the CSP will automatically scale up the resources and process the incoming requests. Gradually, the attacker succeeds in capturing all the resources and the server will run out of resources leaving its legitimate users sub-serviced.

There are mainly two types of DDoS attacks:

- bandwidth attack or network level attack
- application attack or service level attack.

Bandwidth attack is targeted towards the network bandwidth that the service provider uses and floods the entire bandwidth so that the targeted server cannot service its

legitimate users. The attacker floods the target with numerous invalid requests like half-open TCP connections (SYN flood attack). In the service level attack, the attacker floods the target with numerous legitimate-like requests with malicious intent (asking for a service provided by CSP). As the content of the requests looks similar to the ones coming from legitimate users, the CSP won't be able to discriminate it and provide services unnecessarily to such users also. Application attack will be targeted towards the victims' resources like the memory, CPU etc.

Hence, authenticating the legitimates and authorising them to access the resources without any hindrance is the major focus of the proposed approach. This has been achieved in EnEra, the proposed methodology in two phases. In the first phase, the user identity is scrambled in order to protect the service set identifier (SSID) from the intruders. The surviving mechanisms are vulnerable to brute-force attack that could be resolved in a four step process discussed in this paper. Further, the method has been strengthened in the second phase, by validating the trust level of the user and the illegal users are eradicated.

Rest of the paper is organised as Section 2 details about the existing works, Section 3 describes the proposed work followed by experimental setup and result in the analysis in Section 4, performance analysis is detailed in Section 5 and concluded with the future direction in Section 6.

2 Related work

The connection to the private network is secured by using authentication techniques like wireless enabled protocol (WEP), wireless protected access pre-shared key (WPA2) which are mandatorily available on network devices from 2006. But these encryption techniques are vulnerable to brute force attack. The network commands as (Thandeeswaran et al., 2016b).

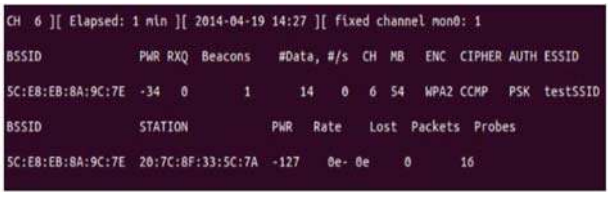
- Airmon-ng

  This script can be used to enable monitor mode on wireless Interfaces. This starts the virtual monitoring on the interface. Airmon-ng start mon0 command starts the monitor mode on the interface of intruders system.

- Airodump-ng

  This script is used for capturing the present wireless network and the security algorithm used. It can also get base service set identifier (BSSID) of the connected system to the victim network as shown in Figure 1.

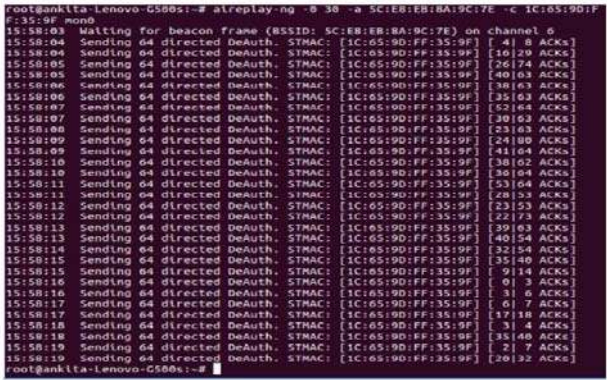
Figure 1 Airodump-ng command output (see online version for colours)



- Aireplay-ng

  This script captures the traffic from the connected system over the victim's network as shown in Figure 2.

Figure 2 Aireplay-ng command output (see online version for colours)



- Airocrack-ng

  Once the traffic has been capturing this script it can compare the captured text with randomly generated text in the file. As shown in Figure 3. After getting the security key the intruder can connect to the network. An intruder can analyse the traffic pattern, firewall, penetrate any vulnerable script in the network, spoof the MAC address and get the access to restricted domains.

Figure 3 Airocrack-ng command output (see online version for colours)

In EnEra, Level 1, the SSID is secured by hiding its broadcasting and then encrypting with secure algorithm using the encryption plug-ins on the propagating network device.

The internet and a web browser are the only requirements for getting the cloud services. The organisations using the cloud service need not be bothered about the

maintenance and management of these resources. The providers can scale up and down the resources delivered on-demand basis. The resources which a cloud provides can be accessed anywhere anytime.

Roberts and Al-Hamdani (2011) has discussed 'wrapper attack and flooding attack' or denial of service (DoS) attack. 'Reputation fate sharing' is another serious issue discussed here. Side channel attack in which the sharing of hardware resources lead to data leakage from one system to another is also a probing threat. Sun et al. (2011) has classified the security issues into six categories. Privacy issues like enabling users to have control over data, preventing data loss while replicating etc are also discussed. Subashini and Kavitha (2011) have stated various security issues in the different delivery models of the cloud. Zissis and Lekkas (2012) have addressed various security issues like trust, confidentiality and privacy, integrity and availability.

Spoofing the IP address of virtual machines is another serious security challenge. The malicious users get the IP address of the virtual machines and implant malicious machines to attack the users of these VMs. This enables hacking and the attackers can confidential data of users and use it for harmful deeds. As the cloud is providing on-demand service and supports multi-tenancy, it is more prone to DDoS attack also. Unless data leakage prevention (DLP) agents are embedded in the cloud, due to multi-tenancy and moving away from data from users control to cloud environment, the problem of data leakage will also be there.

Cloud computing has become a tempting target for cybercrime. The prominent providers like Amazon and Google have mechanisms to defend against this type of attack. But, not all providers do have. Malware injection attacks (Jamil and Zaki, 2011; Liu and Chen, 2010), as in any internet services, are also posing a major threat in security consideration of cloud environment. The attacker will inject malicious codes, services or even virtual machines into a cloud environment.

DPCA, a dual phase authentication, in the first phase, the user is segregated as man or machine. Thereby bot-nets are filtered out and the flooding messages from bots are mitigated. In the second phase, where only man is allowed to access the cloud resources with a hypothetical approach combined with the user intent but not the content. (Thandeeswaran et al., 2016a).

Table 1 Comparison of surviving techniques

Techniques	*Merits*	*Demerits*	*Tools used*
A new trusted and collaborative agent-based approach for ensuring cloud security (Roberts and Al-Hamdani, 2011)	Ensures the security and privacy both at CSP level as well as user level	Not able to find the attackers packet among the incoming packets	Java programming language
Fine-grained capabilities for flooding DDoS defense using client reputations (Natu and Mirkovic, 2007)	Consumes only less memory Operational cost is also reduced Tickets validity is terminated as soon as the server changes its secret key.	Does not address the issue of dynamic addressing Fails if human attacker Clients turn hostile after acquiring ticket	Emulab testbed

Table 1 Comparison of surviving techniques (continued)

Techniques	*Merits*	*Demerits*	*Tools used*
JUST-Google: a search engine-based defense against botnet-based DDoS attacks (Al-Duwairi and Manimaran, 2009)	Reduced traffic aggregation near victim to a great extends Clients with authenticated IP are only given access	Fails when attacker is human Communication overhead	Not implemented
DDoS defense by offense (Walfish et al., 2010)	No client or network modification No communication overheads Attackers can't fool the defense mechanism	Increases cost at server, network as well as end user side Works only if good clients have enough bandwidth Unfair resource allocation No discrimination between DDoS and flash crowd	Emulab testbed
Multi-level authentication technique for accessing cloud (Dinesha and Agrawal, 2012)	Strict authentication and authorisation Breaking of passwords difficult	Will not work if there are no intermediate levels	Not implemented
Virtualised defense and reputation-based trust management (Hwang et al., 2009)	Protect cloud from various aspects of security threats	No implementation evidence	Not implemented

3 Enriching legitimates and eradicating duplicates – proposed approach

EnEra focuses on mitigating, eradicating duplicates and enriching legitimate users in the cloud infrastructure. EnEra has two levels of operation. In the first level, the identity is concealed from the attackers by scrambling the SSID in four different steps. In the second level, the duplicates are eradicated with their credits earned through their behaviour.

3.1 Level 1: eradicating duplicates by scrambling SSID

In the first phase, the user ID is scrambled in four different steps. Encryption of SSID is made to prevent the access of network from intruders. This makes the intruder hard to get the decrypted SSID without knowing the algorithm used and its preshared key. The address space of single IPv4 address can be used to design the network for the system in the company. The network device like routers, virtual router and gateway translate the internet service providers (ISP) IP to private IP using network address translator (NAT)

as per the required number of systems in the company. The access to the private network can be made secure by using WEP, WPA2 authentication and encryption techniques. Some companies even secure their network by using hidden service set identifier (SSID) which does not broadcast the SSID. So that intruder cannot catch the network easily.

All these techniques are vulnerable to brute force attack and some hacking commands. In this paper, the encryption of SSID is made to prevent the access of network from intruders. This makes the intruder hard to get the decrypted SSID without knowing the algorithm used and its preshared key. The network device used for NAT maintains access list of system's with its media access control (MAC) address to allow them to connect to the private network. The addition of SSID encryption and maintaining access list increases the complexity to secure the connectivity in the private network.

It is secured by using strong authentication protocol, the system with decryption plugin can only decrypt the broadcasted SSID and request to the network device for connection. Network device maintains the access list with valid MAC address. At first, an intruder cannot see the hidden SSID easily but if the intruder has got the SSID then she has to send the SSID to the router. But as the decryption algorithm is not known to the intruder it will send encrypted SSID to the network device which will deny the request. The maintenance of MAC address table gives access to valid systems only.

In this paper, there are four phases for securing the connection to the private network shown in Figure 4.

Figure 4 Four security phases (see online version for colours)

1 Hidden SSID

SSID is the name given to the network and it is 32 alphanumeric case sensitive characters set. The systems trying to connect the same WLAN network should use the same SSID as it attached to the header of sent packet and gives an identity. The broadcasting of SSID can be disabled. This makes hard for the intruder to catch the SSID easily.

2 Encrypted SSID

The network device broadcasting hid SSID is having plugins for encryption of SSID with a strong algorithm. The system with the decryption plugins for the same algorithm and secure shared key gets the access to a request for network connection. Intruder gets hidden SSID, which is in the encrypted form. And the decrypted SSID can only connect to the network. Network device denies the access of intruder to the network.

3 Secured password

Using the strong authentication protocol, the connection is restricted to the authenticated systems. This password is shared with the group of people of the company.

Table 2 MAC address table

System	*IP address*	*MAC address*
Sys1	192.168.79.2	00-26-5E-56-F8-7D
Sys2	192.168.79.3	00-27-5E-76-FE-A2
…	…	…
Sys252	192.168.79.252	02-A3-5E-2F-AE-B2

4 MAC address verification

The MAC addresses of the system expected to connect the network are maintaining to the network device as shown in Table 2.

The network device will check if the requesting system's MAC address is the MAC table. Then only the requesting system connects to the network.

3.2 Level 2: enriching legitimates by validating trust level

In the second phase, the users are authenticated depending on the credits. Based on the traffic flow (in the case of network level attack) and next on the interval between consequent service requests (in the case of service level attack). Credits are given to users based on this authentication upon which services are provisioned accordingly.

As shown in Figure 5, the proposed solution protects the CSP resources from the threat of network level as well as service level DDoS attacks. In network level attacks, the target will be flooded with numerous invalid illegitimate requests. Such requests are comparatively easier to discriminate from legitimate requests. But, in service level attack, the attacker floods the target with 'legitimate-like' requests and such request will have all traits similar to that of a legitimate request. It is observed that only the user behaviour can be used as a criterion to detect this kind of attack.

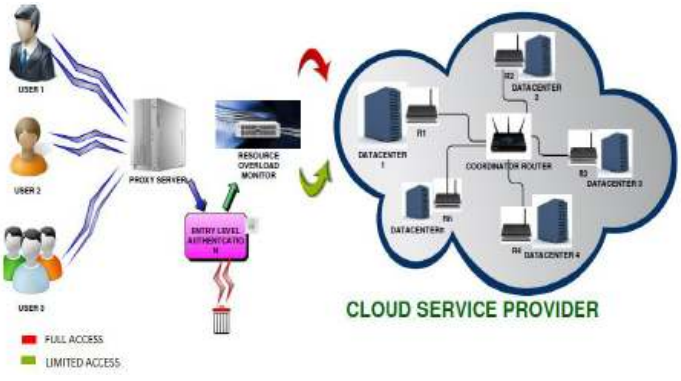
The approach is an authentication based approach that classifies the users into three ranks of reputation, viz. *well-reputed, reputed and ill-reputed*. The users are given credits for this classification. The lower limit of credit is L_VALUE and the upper limit is H_VALUE.

Initially, all users are given a credit equal to M_VALUE which is the mean of L_VALUE & H_VALUE. P_VALUE, a predetermined value (L_VALUE < P_VALUE < H_VALUE) is the deciding factor for reputation. Those users who acquired credit value greater than P_VALUE are designated as *well reputed* and are given full access to CSP resources. The users with credit value between L_VALUE and P_VALUE are given limited access by classifying them under group *reputed*. Others whose credit is less than L_VALUE are blocked and blacklisted.

3.3 Assumptions

- The attackers form botnets by compromising vulnerable systems distributed across the network and installs malicious program codes in those systems. This can happen with or without the knowledge of that system. Whatever the case may, the instructions in such codes will make the systems to send requests to flood the target server. So, it is assumed that these request patterns will exhibit signs of similarity as they are the result of same program code installed in all zombies. This is our assumption used to defend network-level DDoS Attack.
- Again, as it is the same program code or botmaster that triggers all zombies, there will be a periodicity in the inter-arrival time between consequent requests from a user. This is used as an assumption to defend against service level DDoS Attack.

Figure 5 Architecture of network level attack defense mechanism (see online version for colours)



The request from the users will be accepted by a proxy server which does the entry level authentication of the incoming requests. This presents the user with some puzzles to distinguish human users from robots. After this phase, the bad traffic is trashed and others are given to a component called resource overload monitor (ROM). Based on the volume of requests, this will detect whether there is flooding or not. In the case of resource overload, the flow routers at data centre perform flow analysis and give the result to the coordinator router (CR). The CR compares the inputs from all flow router. If the requests with similar contents are valid requests, they are concluded to contribute to service level flooding. Invalid requests contribute to network level flooding. The details of discarded and accepted flows are communicated to the ROM and it will add or deduct credits of users accordingly.

In the case of service level attack, the flow router reports the inter-arrival time between requests from each aggressive users to the CR. The CR discards the requests from users who send requests in fixed intervals. The details of discarded and accepted flows are communicated to the ROM and it will add or deduct credits of users accordingly.

The proposed framework (Figure 5) authenticates the users in three phases

1 *Phase I*: The proposed framework first finds out whether the incoming requests are sent by a human user or programmed robot. This is done by presenting the sender of requests a puzzle or mathematical sum. Only human beings can succeed in crossing this test. So, we can easily confirm that the others who fail in the test are robots with malicious intent. Such senders are blocked and their IP addresses are blacklisted immediately. This can be considered as an entry level authentication.

2 *Phase II*: After authenticating the users based on the above test, the system detects whether there is flooding or not based on the volume of incoming traffic. The system has to analyse the flow similarity. If the flow analysis yields similar results, the system checks the content of requests. The valid requests are given for *Phase III* authentication. The senders of invalid requests with similar content are suspected and those requests are dropped. The credits of such senders are decremented. Also, well-reputed users are notified about the possibility of malware in their system so that they can take necessary actions to rescue their systems and hence avoid further decrements in credits.

   In case if the flow analysis gives different results, that flow can be concluded as coming from different users and those users are legitimate. The credits of such users are incremented and services are provisioned accordingly.

3 *Phase III:* In case, there is no network flooding, the system has to check for service level flooding in which the attackers will flood the system with 'legitimate like' requests. Phase II authentication fails in such cases. Then the system finds the aggressive users first. Non-aggressive users are considered harmless and based on credits they have, they are provisioned services. Aggressive users can be attackers or impatient but legitimates. Such users' credits are decremented first. Then the inter-arrival period between requests is found out. Those who sends requests in the random interval are 'impatient legitimate'. They are given service according to their credits. Others are suspicious clients. Their requests are dropped and credits are further decremented. Well, reputed clients are then given notification. The users whose credits got exhausted are blocked and blacklisted.

3.4 Crediting mechanism

Initially, all clients are assigned a credit value, M_VALUE, which is the mean of L_VALUE and H_VALUE as represented in equation (1),

$$M\_VALUE = \frac{L\_VALUE + H\_VALUE}{2} \tag{1}$$

Under normal circumstances, the credits of all clients are incremented according to the following equation:

$$Credit_{new} = \min\left(incr \times Credit_{old}, HVALUE\right) \tag{2}$$

where *incr* is an increment factor that can be fixed randomly by the CSP.

Under attack, the CSP will experience resource overload and the credits of aggressive clients are reduced. The credit values of such clients are decremented according to following equation:

$$Credit_{new} = \max\left(Credit_{old} - decr \times Credit_{old}, LVALUE\right) \quad (3)$$

where *decr* is a decrement factor fixed by the CSP.

If those clients were already in the well-reputed list, they are notified about the chance of a virus or Trojan attack and the decrement in the credit. Thus, they can take necessary actions to come out of the viral attack and escape from being penalised further. The credits of other clients are incremented as per equation (3). Traffic from such clients is considered to be as flash crowds and is processed for providing the requested service.

When flooding occurs, the proxy server notifies the ROM and the traffic will be distributed to the to the flow routers which are not busy at that instant. The information regarding the state (busy or not) of flow routers will be communicated to the ROM by the CR. Flow router finds inter-arrival time between consequent requests. After discarding the suspicious flow, the CR informs the proxy server about the legitimate clients. The credits of such clients are incremented by the proxy server and their reputation is checked based on which they are assigned path to the data centres in the cloud.

3.5 Virtues of proposed concept

- The method is effective against both networks as well as service level attack.
- Impatient legitimate is also being served based on credits.
- The method does not have to maintain any predefined profiles of traffic or history of communication. The only thing that has to store is the credit and corresponding reputation of each client.
- The credit expiry mechanism does not allow the credits acquired by one client to be inherited by anyone due to dynamic IP address allocation.
- As flow routers do the function of flow analysis and load balancer distributes the tasks to these routers which are not busy at the instant, there won't be any flooding.
- Notification mechanism to well-reputed users about the likelihood of the presence of malicious programs.

4 Experimental setup and analysis

Ankita and Thandeeswaran established the experimental Level 1, setup shown in Figure 6, for the private network. This setup could eliminate the duplicates by scrambling the SSID.

1 Network device like a router, virtual routers, and gateway routes the ISP's IP as well as broadcast the SSID for the network. The device consists of the property to hide the broadcasting of SSID. They have enabled with the encryption plugins. The strong encryption algorithm is used to design the plugin in Java domain. Using this functionality and hidden SSID functionality on the network device allows decrypting the SSID and hiding the broadcasted SSID. These makes harder for an intruder to get hidden SSID and then decrypt without knowledge of encryption algorithm and shared key. The network also maintains the MAC access table with all valid expected system's MAC address.

2 Connected systems: as per the address space allocation of IPV4, the specific number of systems can connect to the network. All the expected system's MAC addresses are listed in the MAC table of the network device. Trusted people in the company can have the encrypted SSID which they use for connection from valid system to the network. These systems are with the decryption algorithm plugin. The system can catch the hidden SSID and decrypt the entered SSID requesting to the network device for connection.

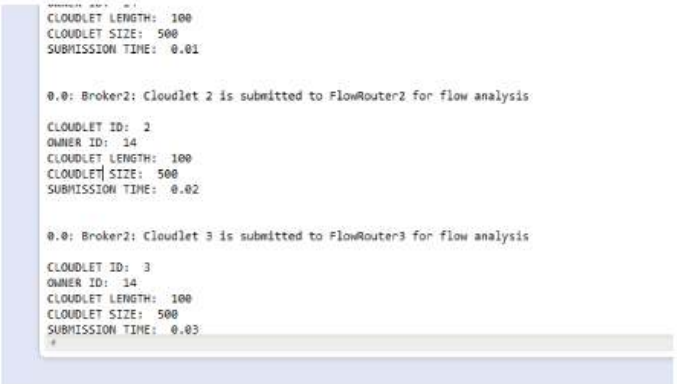
Figure 6 Experimental setup for private network (see online version for colours)

The proposed Level 2 concept has been tested and verified by simulating the cloud environment using CloudSim Toolkit 3.0. Three different scenarios have been simulated, namely, normal scenario, attack scenario before implementing the proposed solution and attack scenario after implementing the proposed solution. It is found that the performance of data centres owned by the CSP remains unaffected even during the attack period after implementing the proposed concept. The flow router analyses the flow and finds out the cloudlet length, cloudlet size and cloudlet submission time.

Figure 7 Non-aggressive users credits incremented (see online version for colours)

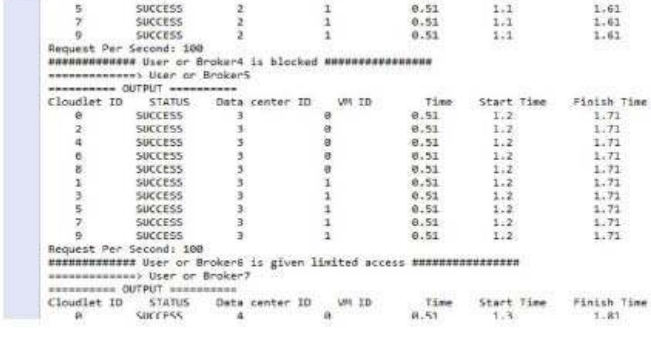
As shown in Figure 7, the credits of non-aggressive users are incremented. Further flow analysis is not performed. They are directly given the requested services. The requests from aggressive users are given for flow analysis as shown in Figure 8.

Figure 8 Performing flow analysis of aggressive user requests (see online version for colours)



The result of flow analysis is submitted to the CR and the CR find the inter-arrival time between requests from each user. The credits are decremented if it is constant and the requests from such users are dropped. If the request arrival interval is random, the credits are decremented for being impatient but are given services according to the credit attained. The reputed and well-reputed users are allowed to access CSP resources whereas others are blocked and blacklisted. The users with credits greater than or equal to 18 (H_VALUE) are given full access to CSP resources. The users who acquire credit value greater than 10 (L_VALUE) but less than 18 (P_VALUE) are given limited access and other users whose credits go below 10 is blocked and blacklisted as shown in Figure 9.

Figure 9 Final output (see online version for colours)

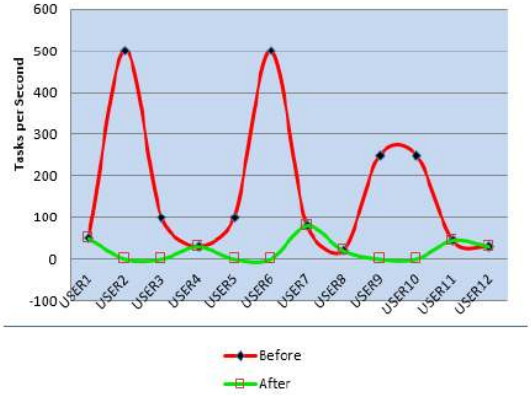


4.1 Performance analysis

The traffic at data centre includes the requests from legitimate users as well as attackers. This will contribute to flooding. The proposed system has completely eliminated the requests from ill-reputed users whereas the well-reputed users are given full access as before.

The graph, Figure 10, depicts the data centre availability before and after deploying EnEra, the proposed methodology. Considerable increase in the number of tasks executed and the rate of data centre access have been observed. The increased spike at the users 2, 6, 9 and 10 denoted that they are attackers and users 3 and 5 are suspicious users. The tasks submitted by these users are completely discarded without disturbing the legitimate clients after employing our method. Thus, the proposed method reduced the flooding at CSP and hence CSP can perform more efficiently even in the case of attack period.

Figure 10 Traffic at data centre before and after the proposed solution (see online version for colours)



Despite the fact that configuring an access point to not allow the beacon frame to include the SSID provides little protection. This may prevent a 'casual' unauthorised user or novice attacker using Windows XP from capturing the SSID and entering the network. On those APs that do allow this configuration, SSID beaconing should be turned off and the SSID entered manually on each device.

4.2 Resource utilisation with respect to user

Resource utilisation here means how much percentage of CSP data centre resources are allotted to each client. This includes the CPU, RAM and bandwidth. As per the proposed method only well-reputed users are given full access to CSP resources, reputed users are given limited access and ill-reputed users are fully blocked.

The graph in Figure 11 depicts that well-reputed users such as users 1, 4, 7,8,11 and 12 are given full access to CSP resources. The users 3 and 5 who are suspicious are given limited access to resources whereas the users 2, 6, 9 and 10 are completely blocked from

accessing the CSP resources. Earlier the users were given random resource allocation due to which well-reputed users also faced inefficient service delivery from CSP.

Figure 11 Resource utilisation based on credits (see online version for colours)

4.3 Processing cost

The processing cost here means the cost incurred at each data centre in processing the requests from all users. The processing cost at each data centre has decreased tremendously after the proposed method has been applied. Instead of giving as much task as possible to one data centre, the load is distributed among the data centres which will, in turn, lessen the response time for serving clients requests. As shown in Figure 12, earlier only data centres1 and 2 does all the processing and other DCs were idle. But, after implementing our proposed solution, all data centres contributed to CSP service delivery and hence helped in enhanced performance and reduced response time.

Figure 12 Processing cost per data centre (see online version for colours)

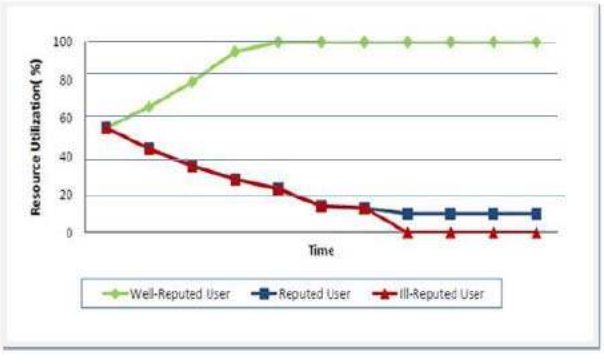
4.4 Traffic at data centre with respect to time

The traffic at data centre is measured by analysing the number of requests reaching the various data centres owned by the CSP in a particular time interval. The request reaching the CSP during a time period of one second is observed here.

4.5 Resource utilisation with respect to time

The resources here refer to the CPU, RAM and bandwidth. The utilisation in percentage (%) according to the credits attained by users is observed. As per the proposed solution, the well-reputed users with higher credits are given full access to these resources. The graph in Figure 13 depicts that the users whose credits increments with time and crosses a certain prefixed limit are allowed to access the 100% CSP resources.

Figure 13 Resource (CPU, RAM and bandwidth) utilisation based on credits (see online version for colours)



The reputed users are allowed to use the CSP resources in a limited manner. Also, the ill-reputed users whose credits got exhausted are totally debarred. The proposed scheme emphasise strict following of this rule so that the users will get serviced based on the reputation and no attacker is being served unnecessarily which was the case before implementation of the proposed approach in a cloud environment.

From the above discussion, it is very clear that the proposed scheme can aid the CSPs to get rid of the hazard of DDoS attack completely. The method is efficient in terms of computational overhead and memory consumption. The communication between the entities consumes time. Even though, owing to the adeptness of the proposed methods to detect and put off the outrage of DDoS in the cloud which handles the critical business of and provides services to a huge community, the communication overhead which may crop up can be ignored. Cloud environment has given 100% protection from the threat of DDoS attack. The method is cost efficient also. It does not demand much change in the existing infrastructure and doesn't involve any complex calculation.

5 Conclusions and future work

The private network gets more secure with the four phases as hidden SSID, encrypted SSID, Secured Password, MAC access table for authenticating the requesting system. The intruder could not get to the network easily. DDoS attack in the cloud is one of the most dangerous security issues that prevail in cloud computing environment. Although numerous researchers' works which have been undergoing in different parts of the globe has come with innovative solutions, none of them proved to be 100% effective in defending this type of attack. This paper proposed a three-phase authentication scheme that helps to discriminate the DDoS traffic from the flash crowd. Credits are given to the authenticated users and users are categorised into various reputation classes based on this credits. Service provisioning depends on the reputation of the user. The method deals with both networks as well as service DDoS. There is no need of maintaining any traffic profile for comparison purpose. Multi-level authentication helps in the more efficient validation of legitimate users. The method doesn't involve any complex computations and memory overhead. The proposed method is expected to detect, discriminate and prevent the DDoS attack and ensure 100% protection from the overwhelming threat of DDoS in the cloud. This defence mechanism will be extended to a federated cloud environment where the CSPs share the credits attained by users in a secure way.

References

Al-Duwairi, B. and Manimaran, G. (2009) 'JUST-Google: a search engine-based defense against botnet-based DDoS attacks', *Proceedings of IEEE International Conference on Communications*, Dresden, Germany, pp.1–5.

Dinesha, H.A. and Agrawal, V.K. (2012) 'Multi-level authentication technique for accessing cloud services', *Proceedings of IEEE International Conference on Computing, Communication and Services*, Uttar Pradesh, India, pp.1–4.

Hwang, K., Kulkarni, S. and Hu, Y. (2009) 'Cloud security with virtualized defense and reputation-based trust management', *Proceedings of Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing*, Chengdu, China, pp.717–722.

Jamil, D. and Zaki, H. (2011) 'Security issues in cloud computing and countermeasures', *International Journal of Engineering Science and Technology*, Vol. 3 No. 4, pp.2672–2676.

Jeyanthi, N., Shabeeb, H., Thandeeswaran, R. and Saleem Durai, M.A. (2014) 'RESCUE: three phase authentication to detect and prevent DDoS attacks in cloud computing environment', *International Journal of Engineering, Transaction B: Applications*, Vol. 27, No. 8, pp.1137–1146.

Liu, S-T. and Chen, Y-M. (2010) 'Retrospective detection of malware attacks by cloud computing', *Proceedings of International Conference on Cyber-Enabled Bi-level User Authentication in Cloud Infrastructure Distributed Computing and Knowledge Discovery*, Vol. 71, pp.511–517, Academic Press, UK.

Natu, M. and Mirkovic, J. (2007) 'Fine-grained capabilities for flooding DDoS defense using client reputations', *Proceedings of ACM Workshop on Large-Scale Attack Defense*, Kyoto, Japan, pp.105–112.

Roberts, J.C. and Al-Hamdani, W. (2011) 'Who can you trust in the cloud? A review of security issues within cloud computing', *Proceedings of ACM Information Security Curriculum Development Conference*, Kennesaw, Georgia, pp.15–19.

Subashini, S. and Kavitha, V. (2011) 'A survey on security issues in service delivery models of cloud computing', *International Journal of Network and Computer Applications, Science Direct*, Vol.34, No.1, pp.1–11.

Sun, D., Chang, G., Sun, L. and Wang, X. (2011) 'Surveying and analyzing security, privacy and trust issues in cloud computing environments', *Advanced in Control Engineering and Information Science, Science Direct*, Vol.15, pp.2852–2856.

Thandeeswaran, R. and Saleem Durai, M.A. (2016a) 'DPCA: dual phase cloud infrastructure authentication', *International Journal of Communication Networks and Information Security*, Vol. 8, No. 3, pp.197–202.

Thandeeswaran, R., Ankita and Jeyanthi, N. (2016b) 'Securing service set identifier of wireless network', *International Journal of Pharmacy and Technology*, Vol. 8, No. 3, pp.16605–16610.

Walfish, M., Vutukuru, M., Balakrishnan, H., Karger, D. and Shenker, S. (2010) 'DDoS defense by offence', *ACM Transactions on Computer Systems*, Vol. 28, No. 1, pp.3:1–3:54.

Zissis, D. and Lekkas, D. (2012) 'Addressing cloud computing security issues', *Future Generation Computer Systems*, Vol.28, No. 3, pp.583–592.