International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2021/209700 Volume 8, Issue 4, July – August (2021)

RESEARCH ARTICLE

A Novel Three Layer Filtering (3L-F) Framework for Prevention of DDoS Attack in Cloud Environment

A. Somasundaram
PG & Research Department of Computer Science, Chikkanna Government Arts College, Tirupur, Tamil Nadu, India.
somasundaram.a@gmail.com

V. S. Meenakshi
PG & Research Department of Computer Science, Chikkanna Government Arts College, Tirupur, Tamil Nadu, India.
meenasri70@yahoo.com

Received: 08 June 2021 / Revised: 01 July 2021 / Accepted: 07 July 2021 / Published: 28 August 2021

Abstract – Data security is an integral requirement of any modern information system as attackers are gaining chances due to the prompt improvement in digital technology. However, in the current decade, the use of cloud computing is rising steeply, and so is network traffic. As the cloud computing model is based on the distributed computing, cloud servers are widely distributed and cloud users can access the service from anywhere and at any time. This makes the cloud servers, a target for the adversaries. The most common attack in a cloud environment is the DDoS attack that causes bulky and abnormal traffic to the cloud server. The cloud server is incapable to manage such unusual traffic and stops momentarily by making the server down with excessive traffic. DDoS attacks can be avoided by diligent traffic control prior to the DDoS attack. This paper proposes a novel three-layer filtering mechanism to prevent various forms of DDoS attacks. The first layer of the proposed DDoS attack prevention mechanism uses two-level authentication processes. Second layer filtering verifies whether the user accesses the resources within the pre-defined limits and the third layer filtering sieves out the spoofed packets. The proposed model has been analyzed for evaluating the performance in terms of CPU overhead and load, the throughput of the victim, the reduction in connection delay. The result analysis shows that the proposed model has improved performance with a higher detection rate of 0.92 and a lower dropout rate of 0.10.

Index Terms – DDoS Attack, Cloud Computing, Cloud Security, Attack Prevention and Cloud Server.

1. INTRODUCTION

In a cloud environment, various operations such as storing, maintaining, controlling and accessing the data are performed more frequently than the operations performed on a local server and personal computer. Cloud computing through its potential, flexibility and cost minimization characteristics offers the proficiency to share resources in a persistent and apparent way; besides it can carry out processes that resolve unique desires. Furthermore, cloud computing gives on-call offerings to its consumers and allows them to access the common infrastructure. Cloud offerings have become increasingly popular in various public and business firms.

One of the major vulnerabilities targeting the accessibility of services in the cloud computing world is the Distributed Denial of Service (DDoS) attack. In general, the DDoS attack prevents genuine cloud users from obtaining the services or resources delivered by cloud suppliers [1]. This is done by draining the server's computing power by flooding the network bandwidth, which ultimately leads to a lack of access to cloud services or resources, ending with subsequent financial losses [2]. The DDoS attack is a significant security concern to be solved in a cloud computing environment where many users share its resources based on their demand. DDoS attacks differ in complexity and scale in which an invader can make a forged request appear like arbitrary garbage on the network, or in an additional problematic way by making the attack traffic resembles real web traffic.

The DDoS attack is intended to bring the target cloud down. Habitually, the unspoken goal of this attack is to limit accessible resources and disrupt the victim cloud's claimed services and thus the victims are harassed as a result of significant economic loss [3]. The assailant compromises the victim's concealment and accesses their sensitive data for various malevolent activities. Besides, gaining recognition in the hacker community is an optimistic justification for these outbreaks. In the cloud environment, all spiteful attackers act as intruders for sending an enormous amount of spiteful packets unswervingly to the target cloud servers [4,5]. As a consequence, the whole network gets drowned with attack messages rather than valid packets. Since the attack packets being flooded with cloud storage servers, the obtainability of cloud storage servers for registered consumers will be fictitious. Attackers could also be able to exploit the content of legitimate packets. Meanwhile, it will disrupt the operations of a victim cloud server. Additionally, the consumer may have restricted bandwidth and computing

RESEARCH ARTICLE

resources in the cloud environment [6]. However, any interruption in service can result in customer dissatisfaction. As a result, distinguishing and thwarting DDoS attacks in a cloud background requires a robust and efficient technique. In general, the prevention mechanism offers secured facilities and interrupts the malevolent cause that generates a precarious condition of attack.

Thus, the method has been introduced that uses three layers filtering (3L-F) mechanism for the effective prevention of various forms of DDoS attack. The main aim of the proposed model is to optimize machine resources and system functionality. The proposed work focuses on countering DDoS attacks, which is increasingly difficult with the vast resources and techniques available to attackers. In the proposed model, the first layer filters the unauthenticated users, the second layer filters the authenticated user accessing for more resources and the third layer filters out the spoofed packets and finally results in legitimate requests to the cloud server. Various performance analyses have been accomplished to show the effectiveness of the recommended work. As the proposed model combines various filtering strategies, it is highly effective in the current scenario. Upon implementation in a cloud background, the model reduces the risk of a DDoS attack.

The rest of the paper is systematized as follows. Section 2 reviews the various models or solutions related to the problem definition. Section 3 explains the proposed model with the problem statement and the scheme utilized. In section 4, the steps for the proposed algorithm and the overall approach to prevent the DDoS attack are presented. Section 5 describes the performance analysis of the proposed 3L-F approach and compares the model with the existing methods. Finally, the paper concludes the proposed work and lists out the future enhancements to be made.

2. RELATED WORK

Several DDoS detection and prevention models exist in the literature that helps in developing an efficient defense mechanism for DDoS attacks. These models have made use of various factors as their primary focus such as auto-scaling decisions [7], resource scaling [9], filtering mechanism [8], multi-layer mitigation and safety in creating a defence system [10,11]. These principles include clear guidance on the development of successful and comprehensive solution necessities to assist the cyber security research community in the design of defence mechanisms. Realistic protection strategies in the cloud environment towards various forms of DDoS attack was suggested by examining the influence of DDoS attacks on cloud possessions [12]. As the cloud infrastructure infer different characteristics and behaviours when compared to the conventional networks, the techniques for preventing or stopping DDoS attacks in the cloud also vary greatly from that of conventional networks.

A third-party auditor (TPA) based packet traceback approach that practices Weibull distribution for examining the DDoS attack packets and their source was suggested [13]. The method is effective due to its strong recognition factor that is determined by the weakness left by the intruder. To reduce the cloud user's overhead, it analyzes the traffic flow pattern to produce an attack alert for various cloud customers. However, the method is more complex and thus it consumes more time to process the requests. A framework for packet monitoring using standard hop count filtering (HCF) algorithm in cloud platforms to avoid DDoS attacks was suggested [14]. This approach eliminates the unavailability of cloud resources to legitimate customers thereby decreasing the number of updates, and computation time. However, the detection rate and false negative rate of this model is not analysed.

A host based intrusion security scheme over hypervisor environments for preventing a DDoS attack in a cloud setting was introduced [15]. It employs a key element examination and linear discriminate exploration, as well as a fusion, nature-inspired meta heuristics procedure called Ant Lion optimization. An Identifier/location separation was also used in preventing DDoS attacks [16]. With this method, the attacker cannot locate susceptible hosts by explicitly distributing packets to the hosts. Conversely, the drawback of these methods is that the attacker can still direct packets to hosts that provide services being processed by the server.

The deployment of the Bloom filter was suggested in preserving a DDoS attack [17].The Bloom filter method is a probabilistic data arrangement for association query that yields either true or false by consuming little memory to accumulate evidence on enormous data. But the main drawback of this model is the high CPU overhead. Similarly, a model that utilizes various categorizations for the identification and investigation of synchronous and non-synchronous traffic flow with network surveillance in time-slot was introduced [18].Apart from these methods, several other methods utilize various machine learning models for detecting DDoS attacks [11]. A model that uses ensemble methods with multi-filter feature selection was suggested [19]. Entropy based models gain more attention in recent days in detecting DDoS attacks [20, 21]. Yet, these models increase complexity and overhead upon executing them.

Several standard models and security policies were introduced in the literature to prevent some specific types of DDoS attacks. A prevention model that makes use of a pushback mechanism and guidelines for allocating the resources was suggested to prevent and mitigate DDoS attacks [22]. It concentrates only on bandwidth and resource consumption attacks. A simple model that utilizes distance based metrics were suggested to perceive and stop DDoS attacks and the scheme is specifically suggested for preventing flooding attacks [23]. A model termed C2DF was introduced in which

RESEARCH ARTICLE

employs neural network technique to train the model for effective detection of DDoS attacks [24].It focuses only on TCP attacks. A model that uses a graphical truing test and constraint based approach along with queuing model has been introduced for preventing and detecting DDoS attacks [25].The model missed various parameters to be considered for effective results such as device analysis.

Though several methods are suggested in the literature for preventing DDoS assaults, most of the methods can withstand merely high-traffic DDoS attacks. Many of the approaches offer low accuracy in detecting attacks and consumes more time to provide the results which is the major concern in any prevention system. Some methods may not be effective in utilizing the resources. Besides, complexity is another major concern to be focused on since simple methods can provide better accurate results in a minimum time. Thus, the objective of the proposed work is to overcome the issues and limitations of the existing models by introducing the new model that uses three-layer filtering for effective discovery of DDoS attacks that increases the accuracy rate and decreases the execution time and CPU overhead.

3. PROPOSED DDOS ATTACK PREVENTION MODEL

3.1. Overall Design of the Prevention Model

The primary motive of the proposed layer based filtering approach is to find the flooding of the DDoS attack initiated by the attacker against the cloud target server. The main difficulty exists in providing successful protection to the cloud environment against DDoS attack is distinguishing the legitimate traffic from attack traffic that has been instigated separately. Attackers typically use several data packets having a variety of spoofed IP addresses to outbreak the target, which consumes more resources to search each data packet [26]. However, in the event of a DDoS attack, a large amount of data may come from legitimate hosts and if the source IP has been forged, then the data would be the same in most instances [27].

The most noticeable indication of a DDoS attack is that the site or service unexpectedly becomes unavailable or inaccessible to its user. Such legitimate traffic with a massive increase in incoming requests causes various performance issues and thus for which further analysis is typically needed. In order to solve the issues, many existing methods make use of TTL or hop counts that help in grouping data packets as trustworthy or malicious. The overall design or workflow structure of the proposed DDoS prevention model is described in Figure 1.

The proposed work introduces the three-layer filtering (3L-F) mechanism to thwart the DDoS attack in a cloud background. Instead of having a single security layer to filter the packets, the proposed model suggests three layers namely layer-1 filter, layer-2 filter and layer-3 filter for filtering the incoming data packets which creates a strong shield of defense for the cloud against DDoS attacks. Here, layer-1 filtering deals with user authentication which is a primary level of verifying the identity of the user or the device.

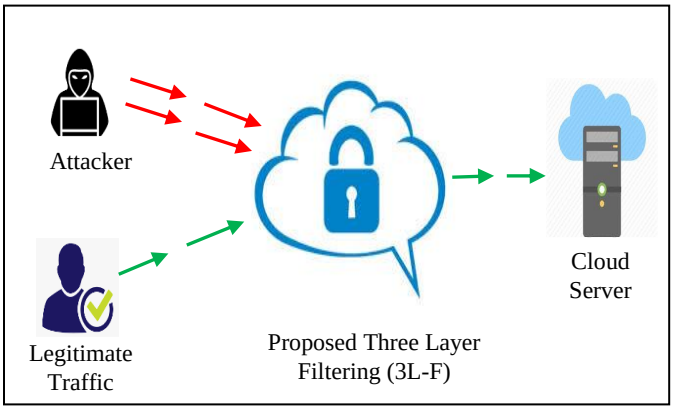

Figure 1 Overall Design of the Proposed DDoS Prevention Model

It uses a two-factor authentication in which apart from normal password verification, it also utilizes a challenge-response authentication model as a second factor. This layer is highly significant since the cloud requires a secure connection for sending the authentication code. The layer-2 filtering is responsible for comparing the quantity of requests that arrived currently with a predefined limit for filtering the data packets. The layer-3 filtering is accountable for identifying the spoofed packets from the legitimate packets. This three-layer filtering based defense model highly increases the detection rate of DDoS attacks with increased performance.

3.2. Design Goals

This section presents the design objectives to be considered in implementing the proposed model. The objectives are consolidated concerning the developers of a comprehensive DDoS defence using a coherent collection of counter-measures. First of all, the designers of DDoS defence systems should strive hard to create a low-cost solution. High dollar costs would prevent widespread adoption of the system by ISPs and possible victims of the DDoS attack [28]. Secondly, it must be possible to understand and confirm the response of the preventive mechanism by the victim. When mitigation is authorized, generally there exists a possibility of mistaking a legitimate packet for a malicious one. In a situation where the victim can process all site visitors notwithstanding ongoing malicious interest, it is probably high-quality to take away false positives by disabling such counter-measures. When they are routed to the server, packets from attackers and valid customers converge to the same subnet and might emerge as sharing hyperlinks [29]. As an effect, protection mechanisms ought to be both upstream filtering and powerful downstream packet discrimination.

RESEARCH ARTICLE

3.3. Proposed Three Layer Filtering Mechanism

The proposed three-layer filtering (3L-F) mechanism has been designed in such a way to achieve the design objective in inhibiting the DDoS occurrence on the cloud atmosphere that makes the service inaccessible to its legitimate users in a highly distributed manner by giving the illusion of legitimate traffic. As the number of attacks and the amount of traffic associated with the attacks continues to rise significantly, there is an immediate need to build a solution to address the problem of DDoS attack recognition and deterrence.

The overall framework of the three-layer filtering mechanism to prevent DDoS attacks is presented in Figure 2. The Three Layer Filtering (3L-F) is a distributed responsive defense mechanism for DDoS attacks in which the Layer-1 filtering authorizes the users and confirms that no hacker is using cloud services. In this layer, the whole process of authentication is processed in such a way that every incoming information or value will be compared with the existing information obtained during the registration phase (profile registered details). Layer-2 filtering is used to restrict the user from gaining access to extra services/resources. This filter ensures that no one can send additional requests to generate flooding attacks knowingly or unknowingly. Layer-3 filtering utilizes the hop count filtering algorithm with some modifications to ensure that no spoofed packet will be entertained. Hop Count Filter (HCF) algorithm is used to filter the IP packets, which requires continuous monitoring during travel over the cloud networks along with a synchronous flag, TTL (Time to Live) and source IP from these packets.

The DDoS threat is initiated by an attacker who sends a massive amount of traffic to the target with the support of agents who block the target's network bandwidth based on IP traffic. The target machine is then subjected to inundated network bandwidth and drops down promptly inhibiting legitimate traffic from retrieving the network. To prevent the DDoS attack, the proposed mechanism is designed with three layers of filtering to select only the legitimate incoming packets. For performing filtering at each layer, various variables are used and are stored in the tables for quick processing. The tables used at various layers for agile filtering is enumerated in Table 1.

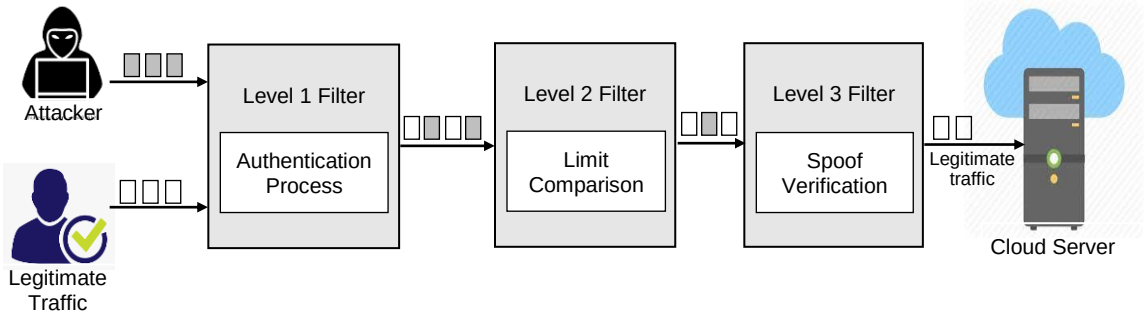

Figure 2 DDoS Attack Prevention with Proposed Algorithm

Table Name	Description
User_profile	Stores all information of the registered user (whole profile)
User_device	Stores information of device and browser used by users
User_auth	Stores user ID, password and Registered Mobile Number (RMN)
Req_details	Stores the information of requests limit and demanding
Pac_details	Stores Source IP Address (TSIP) of packets, Hop-Count for that IP Address (HS) and Synchronous Flag (SF)

Table 1 Tables Used in the Proposed 3L-F Model

The variables used in the proposed model is presented below.

L_UID - User ID entered by the user during the login process

R_UID - Stored User ID during the registration phase

L_PWD - password entered by the user during the login process

R_PWD - Stored password during the registration phase

OTC - One time challenge involving the mathematical operation

OTR - One time response for the given challenge

L_UDev - User device currently used

L_UBrow - Browser currently used by the user

H_UDev - User device history

H_UBrow - Browser history of the user

L_Req - Predefined limit of requests for the specific source

N_Req - Number of requests received from the specific user

RESEARCH ARTICLE

SF- Synchronous flag bit extracted from TCP/IP packet

N_pac - Number of incoming packets from various sources

L_pac - Predefined limit of incoming packets from various sources

SIP - Source IP Address extracted from TCP/IP packet

TTL - Time-to-Leave value extracted from TCP/IP packet

TF - Final value of TTL (Time to Live)

TI - Initial value of TTL

HC - Computed Hop Count value (TI-TF)

HS - Stored Hop Count value

At each layer, the information extracted from the users' request is compared with the attributes of one or more tables for filtering the packets at each level. The Algorithm1 presents the proposed Three Layer Filtering (3L-F).

Input: Users' credentials such as L_UID,

Output: Acceptance or Rejection of packets

Begin

//Layer-1 Filtering

Step 1. Compare the entered login credentials with that of the registered one along with the one-time challenge-response authentication.

If authentication is not successful

Deny the service and Exit

End If

//Layer-2 Filtering

Step 2. Compare the number of received requests demanded by the user with the Predefined limit of requests

If the number of requests exceeds the limit

Deny the service and Exit

End If

//Layer-3 Filtering

Step 3. Verify the Packets based on Hop-Count

If the packet is spoofed

Discard packet;

End if

Step 4. Repeat steps 2 & 3 until the user logout

End algorithm

Algorithm 1Three Layer Filtering (3L-F)

The filtering process at each layer is briefly described in which the initial phase is the layer-1 filtering for authenticating the user which is essential for any application to verify the users' identity. The users' credentials entered at the time of the login process such as user name, password (L_UID, L_PWD) is initially compared with the registered users' credentials (R_UID, R_PWD). Along with the user credentials, the onetime challenge-response are further more utilized to authenticate the consumer is not the bot. In the offered model, the simple mathematical operations involving small numbers are sent as a challenge to the user's registered mobile number.

The user must provide the answer for the given challenge as a response along with the user name and password in the cloud application. The device details and the Brower details are also collected during the user registration and the details are updated frequently. If the browser or the device used to log in to the application is different from the details stored previously, then the complexity of the challenge will get increased with large numbers, or else, the challenge given to the user will be simple.

Thus the first layer filtering authenticates the user by using challenge-response based user authentication even if the user uses a different device or browser for accessing the service with varying levels of complexity. This filter confirms that no hacker is using cloud services. In the whole process of authentication, every current value will be compared to the history of values stored in the tables.

In the proposed work, three tables such as User_profile, User_device, User_auth are employed to store all the information about the user, device and browser used by the user, and user credentials including user name, password and registered mobile number respectively when the user registers their details with the application. The Algorithm 2 represents the user authentication at Layer-1 filtering is shown below.

Input: User's credentials and device details (L_UID, L_PWD, L_UDev, L_UBrow)

Output: Authentication Status

Begin

Compare entered users' credentials with the credentials (R_UID, R_PWD) stored in the User_auth table

If (L_UID == R_UID &&L_PWD == R_PWD)

Compare device and browser used by the user

currently with the history of device and browser details

(H_UDev, H_UBrow) used by the user from the

User_device table.

If (L_UDev == H_UDev&&L_UBrow == H_UBrow)

RESEARCH ARTICLE

```
        Generate simple mathematical challenge
        Send the challenge to the User's RMN
        Verify the response (OTR) of the user for the given
         challenge (OTC)
            If received_OTR == OTR
               Auth_status = success
            Else Auth_status = failure
    Else
        Add the details about the device and browser in the
        User_device table
        Generate complex mathematical challenge
        Send the challenge to the User's RMN
        Verify the response (OTR) of the user for the given
         challenge (OTC)
        If received_OTR == OTR
           Auth_status = success
        Else Auth_status = failure
Else if login credentials are unavailable
    Register the user details
If Auth_status == success
    Allow requests and compare the demanding request with
     a pre-defined request.
Else Deny the service and exit
End algorithm
```

Algorithm 2 User Authentication at Layer-1 Filtering

Once the user is authenticated successfully, then the next step is to filter the number of packets to be serviced for each user at a particular point of time to make resources available for its other legitimate users to provide service. The second phase is the layer-2 filter and it restricts the user from gaining access to extra services/resources. This filter ensures that no one can send extra floods knowingly or unknowingly. The filter makes use of the Req_details table which stores the information about the limits on the number of requests to be processed for all the users and updating the current request field from time to time. In the proposed method, if the request packet originating from the same IP address is more than 25 requests per second, then the source is considered to be the illegitimate bot and thus the IP address gets blocked [30]. Thus, the server will be safe from flooding attacks. The Algorithm 3for layer-2 filtering is given below.

```
Input: Number of Incoming requests
Output: Allow/disallow source packets
Begin
Count the number of requests from the particular IP address
(N_Req) and update the details in the Req_details table
Compare N_Req with L_Req
   If (N_Req>L_Req)
      Deny access to the requested resource
      Revoke the resources allocated previously
      Exit the process
   Else
      Allow request packets and check the packets
   End If
End algorithm
```

Algorithm 3 Layer-2 Filtering

The third phase which is layer-3 filtering makes use of a modified basic Hop-Count algorithm for effective detection of DDoS attacks by filtering spoofed packets. In the proposed algorithm, there was a requirement of using four cases to recognize the legitimate packets however the existing algorithm uses only two cases to identify the attack packets. This makes the algorithm stronger than the existing hop count algorithm [31]. The proposed algorithm ensures that no spoofed packet will be entertained. Hop Count Filter (HCF) algorithm is used to filter the IP packets, which requires continuous monitoring during travel over the cloud networks and extract synchronous flag, Time to Live (TTL) and source IP from the packets. It recognizes two cases for each captured packet in the entire process. Generally, this layer uses the Pac_details table that stores Source IP Address (SIP) of packets, HopCount for that IP Address (HS) and Synchronous Flag (SF) for effective analysis.

In general, Time To Live (TTL) is an 8-bit field in a packet demonstrating the determined lifespan of an IP packet. While the packet is transmitted from sender to terminus, the packet is allowed to move across various routers in the network. Upon receiving the packet, each router diminutions the TTL value of an IP packet by one. The server stores the TTL value of the packets in the IP2HC mapping table however in the proposed algorithm the values are stored in the Pac_details table. Obviously, the destination checks the TTL values of the received IP packet with that of the one stored in the Pac_details table which is computed by the Hop count filtering algorithm [32]. The Algorithm 4 for layer-3 filtering to verify spoofed packets is given below.

```
Input: Incoming requests
Output: Allow/disallow source packets
Begin
Set TTL and SIP values from TCP/IP packet
```

RESEARCH ARTICLE

```
Check the packets for Synchronous Flag (SF) is set
  If (SF ≠ 1)
    Discard packet (spoofed packet)
  Else
    Check the number of incoming packets is limited
    If (N_pac>L_pac)
      Discard packet (spoofed packet)
    Else
      Check the value of SIP is set
      If (SIP≠ 1)
        Discard packet (spoofed packet)
      Else
        Compute the Hop-Count (HC) of the received
        packet
        HC=TI-TF // Hop-Count Value
        Compare HC with HS (Stored Hop-Count)
          If (HC ≠ HS)
            Discard packet (spoofed packet)
          Else
            Update the details in Pac_details
            Allow packets (Legitimate packets)
          Endif
      Endif
    Endif
  Endif
End algorithm
```

Algorithm 4 Layer-3 Filtering to Verify Spoofed Packets

Initially, the packets are verified that whether the synchronous flag is set for the incoming packets. If it is not set, then the packets are considered spoofed and are discarded. Then the number of incoming packets is verified in such a way that the number of incoming packets from different sources exceeds the specified limit then the packets are considered as the spoofed packets. In the proposed algorithm the limit for the number of packets is taken as 40 packets per second (L_pac). Then the source IP address (SIP) is verified to check whether it is set in the incoming packets. If the SIP is not set, then the packets are considered spoofed and are discarded. Finally, the hop count of the incoming packets is verified with that of the stored ones. If there is a mismatch in the values, the packets are considered spoofed and are discarded. However, if the packets withstand the verification process, then it is considered as the legitimate packets and is allowed to processing further.

Cloud Service Provider (CSP) has registered cloud user's mobile number with cloud services. These registered mobile numbers are maintained in the table by CSP and are used to authenticate the users with the help of registered mobile numbers by providing a secure connection [17, 18]. This process has two cases in which the first case is at the time of creating a user ID and Password in which user authentication utilizes a secure connection to provide the secret unique key to the cloud user based on their registered mobile numbers such that the secret key is not tempered by a malicious person.

After successful completion of the first process, the authentication process also creates User_profile, User_device, User_auth tables for maintaining user information, history of devices and browsers as well as credential details used by the corresponding users and handovers the copies of both the tables to the CSP [33]. The second case is at the time of using a new device or browser in which if a cloud user uses a new device or new browser for accessing cloud services then again it would be authenticated with the help of a secret unique key or by asking security questions. Upon successful completion, the User_device table will be updated by adding new information on the device or browser. In the whole process of authentication and both cases, if the cloud user is unsuccessful then access will be denied. In other words, any unauthenticated person will not be able to use or access cloud services.

With the proposed authentication process, an attacker cannot successfully implement the flooding attack. If an attacker is successful in identifying the victim and legitimate user's machine/system, due to the mistaken disclosure of the authentication credentials, the attacker would be able to send only limited requests in the form of flood to the cloud network. This shows that the second filter intended for service limitation will stop the extra flood from entering the cloud network. Unfortunately, still, there is a possibility of receiving spoofed packets by the cloud networks.

In such a case, the third layer filtering with a modified Hop-count algorithm will discard these spoofed packets from the cloud networks and achieve the objective of keeping the server free from DDoS attacks. Thus the proposed mechanism will ensure that the server is utilized only for providing the services to legitimate cloud users. In the authentication process, the model limits the user from accessing the cloud services. Thus, if the attacker passes the first filter, then the second filter will act as a shield by fixing a limit to the requests and even if an attacker sends the malicious packets within the limit then the third filter that utilizes a modified hop count algorithm prohibits the access of cloud resources.

RESEARCH ARTICLE

4. EXPERIMENTAL ANALYSIS

This section presents the results obtained for various performance evaluations made for the proposed work. It includes the planning of group infrastructure architectures and setup to test the proposed DDoS solution. A private cloud network was set up as a testbed using Oracle VM Virtual Box as a virtual environment. The proposed is implemented in python on the private cloud server. Various hardware and software are used in the experimental analysis are Cisco 4000 ISR Series Routers and Cisco Nexus 5000 Series Switch for routing and switching, HP DL-360G8 1U-Rackmount Servers with Intel E5, 128 GB DDR3, 32 TB SSD, Front-end web portal with.NET that supports 2-factor authentication, back-end database with Microsoft SLQ Datacenter and DDoS tools for attack simulation including Low and High Orbit Ion Canon, Packet Storm, Are You Dead Yet (R.U.D.Y), and So on.

The networks are subjected to group and alertness layer attacks using ICMP flooding and a thousand echo requests with buffer sizes ranging from 3700 to 3805 bytes. The use of simulated DDoS attacks denied legitimate users access to the web software portal. When simulating DDoS attacks, real-world user observing archives are used as standards, and arguments for logs have been collected to aid in the generation of DDoS attack graphs. These parameters are chosen because they specify the performance issues experienced by the actual users on the web at any given moment during an attack.

The proposed algorithm is composed of three different defence layers to enhance the security of the cloud environment by preventing DDoS attacks. The proposed work uses three-layer filtering in which each filter uses strong defence mechanisms a shield to stop the attacker from sending the flood to exhaust/exploit the server or compromise the resources. Upon passing the packets through these three levels of filtering, the intruder will seldom be able to intrude into the cloud server. Various traffic parameters are to be considered in implementing the proposed model and are listed in Table 2.

Parameter	**Value**
Genuine traffic category	HTTP
Web traffic method	Web caching model
Genuine packet dimension	584 bytes
Traffic creation at attackers	UDP traffic
Traffic arrival process at attackers	CBR
Mean attack rate per attack host	02 Mbps
Attack packet dimension	584 bytes

Table 2 Traffic Parameters

4.1. CPU Overhead and Load

It is found that the CPU overhead in the FTP server is minimal under the baseline scenario. As a consequence, the server is unable to respond effectively to valid requests [35]. In this analysis, the CPU overhead of the FTP server is estimated and the values obtained for the proposed system and the existing systems such as standard HCF [14] and bloom filter [17] are shown in Table 3. The trial is accomplished by varying the quantity of requests and the obtained results clearly state that there will be an increase in value of more than two-thirds of the peak value when the server is under DDoS attack relative to the baseline scenario. The values presented in Table 3 are depicted as a graph in Figure 3.

No. of Request (sec.)	Existing Mechanism		Proposed Mechanism (3L-F)(%)
	Standard HCF(%)	Bloom Filter(%)	
200	96	98	90
400	94	97	87
600	92	95	85
800	90	93	82
1000	88	91	80

Table 3 CPU Overhead and Load

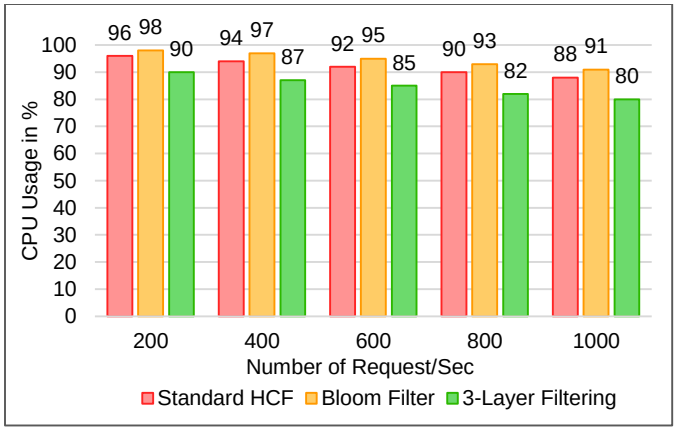

Figure 3 CPU Overhead in Flooding of Request

The results show that the proposed 3L-F mechanism takes less CPU overhead than the existing standard HCF algorithm and bloom filter techniques.

4.2. Throughput of Victim

Throughput is an actual measure of how much data is successfully transmitted from source to destination, whereas bandwidth is a conceptual measure of how much data can be distributed from source to destination [16]. The throughput

RESEARCH ARTICLE

tests the speed while the bandwidth is only indirectly connected to the speed. In general, bandwidth or power is the amount of data that can pass through the link and is measured in bits per second (bps).The throughput is measured based on the attack rate. Thus the experimental analysis has been performed for the proposed model and the existing models such as standard HCF [14] and bloom filter [17] by computing the throughput by varying the attack rate. The results show that when the attack rate is increased, the throughput of the suggested 3L-F procedure is increased in comparison with the values of existing standard HCF and bloom filter models. The obtained results for methods are presented in Table 4. The values presented in Table 4 are presented as a graph in Figure 4 the results and the discussions should be made.

Attack rate (packets/sec)	Existing Mechanism		Proposed Mechanism (3L-F) (Mbps/sec.)
	Standard HCF (Mbps/sec.)	Bloom Filter (Mbps/sec.)	
100	5.5	5.2	5.9
200	4.2	4.9	5.8
500	3.3	4.3	5.7
1000	2.7	2.5	5.5

Table 4 Throughput of Victim Analysis

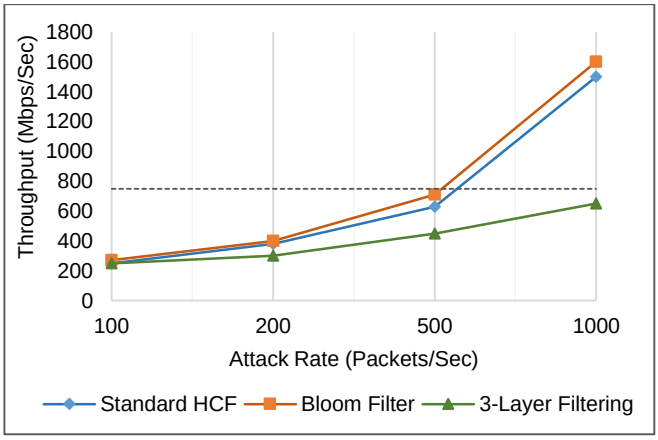

Figure 4 Throughput of Victim

4.3. Reduction of Connection Delay

The next parameter used in the study is the connection delay. In the proposed work, the connection delay has been reduced by deploying three-layer filtering (3L-F) mechanism at edge routers. The result for the analysis is presented in Table 5 by reporting the connection delay in ms by varying the attack rates. The values reported in Table 5 are also presented as a graph in Figure 5 to show the reduction of standard connection delay under dissimilar deployment of existing schemes. In the graph, the connection delay is analysed by varying the attack rate for the proposed model and the existing models such as standard HCF [14] and bloom filter [17]. The results show that the proposed model has a minimum connection delay than the existing models under comparison.

Attack rate (packets/sec)	Existing Mechanism		Proposed Three Layer Filtering (ms)
	Standard HCF (ms)	Bloom Filter(ms)	
100	250	270	250
200	380	400	300
500	630	710	450
1000	1500	1600	650

Table 5 Connection Delay Analysis

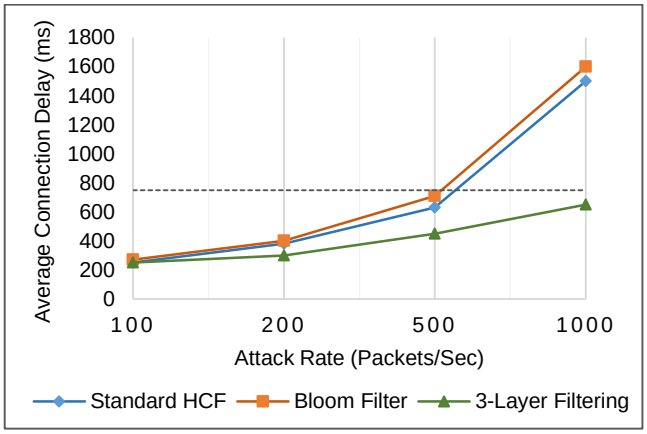

Figure 5 Reduction of Connection Delay

4.4. Detection Rate and Dropout Rate

The detection rate is most significant for analysing the performance of any attack detection model. The experiment has been made with a server and 15 clients to produce the network data traffic that includes both legitimate and attack requests. To generate the DDoS attack traffic, the Netwag tool [34] has been employed. The tool generates TCP SYN attack, smurf attack and so on. Also, the header information in the incoming packets is analysed using JPCap [8]. The experiment is performed with 100 packets in which at each iteration the number of attack packets is increased. Centered on the outcomes obtained from the trials, the performance of the proposed model is assessed with other existing approaches including pushback ([22], distance based [23], C2DF [8], and constraint based [25]. The results obtained from the experiments are investigated and the evaluation is presented as a graph for detection rate in Figure 6, the false negative rate in Figure 7 and the dropout rate in Figure 8.

RESEARCH ARTICLE

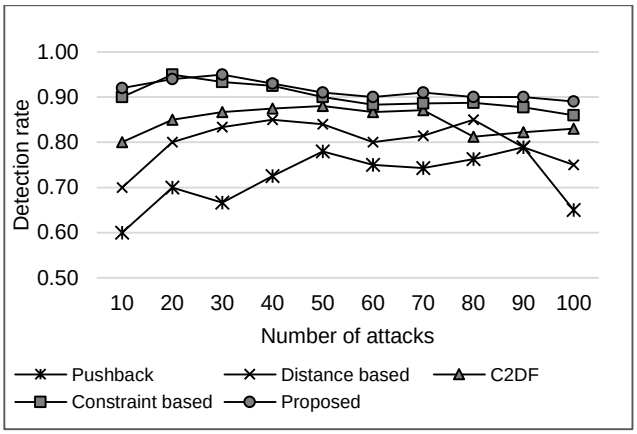

Figure 6 Performance Comparison on Detection Rate

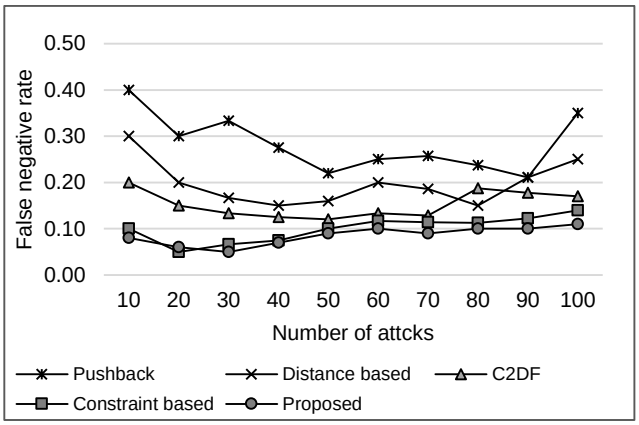

Figure 7 Performance Comparison on False Negative Rate

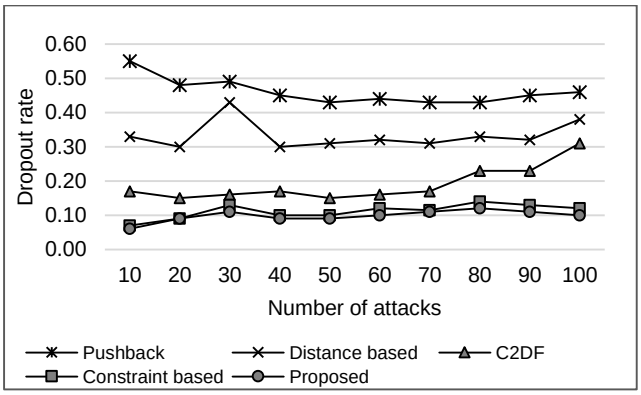

Figure 8 Performance Comparison on Dropout Rate

From the analysis, it is clear that the attack detection rate of the proposed model is better in many cases than other existing methods under comparison including pushback, distance based, C2DF and constraint based models. The average detection rate of the proposed model is 92% whereas the other methods such as pushback, distance based, C2DF and constraint based has the detection rate of 72%, 80%, 85% and 90% respectively. False negative rate specifies the number of legitimate packets that are considered as an attack. In analyzing the false negative rate, the proposed model has a minimum value of 9% than other methods such as pushback, distance based, C2DF and constraint based with a false negative rate of 28%, 20%, 15% and 10% respectively. The dropout rate determines the rate of attack packets left unseen. The proposed model has a minimum dropout rate of 0.10 when compared to the prior methods such as pushback, distance based model, C2DF and constraint based model with a dropout rate of 0.46, 0.33, 0.19 and 0.11 respectively. In general, the pushback takes more time to provide the result whereas distance based and C2DF methods are not effective in resource utilization and accuracy. Though the constraint based method offers better results, the attack detection rate is less when paralleled with the proposed model. Thus, the average CPU overhead of the proposed model is 84.8%, whereas the average throughput and the average connection delay are 5.725 Mbps/sec. and 412.5 ms respectively. Also, the detection rate, false negative rate and dropout rate for the proposed model are 92%, 9% and 10% respectively. The above results analysis shows that the performance of the proposed three-layer filtering mechanism is much better than the existing methods used for the study.

5. CONCLUSION

The DDoS attack is one of the significant threats to the safety of the cloud based environment. The proposed three-layer filtering (3L-F) mechanism is introduced to perform secure prevention of DDoS attacks in the cloud platform by ensuring security. Layer-1 filter deals with effective user authentication because it wants a secure connection for sending the authentication code which then compares the current requests with predefined limits of requests in the layer-2 filter. Finally, the spoofed packets are filtered using layer-3. When a DDoS attack occurs, the proposed prevention algorithm ensures that no unauthenticated person can use the systems; malicious persons cannot send the extra request or cannot demand extra services beyond their limits assigned and so the probability to send the flood is very low implies that all the malicious packets will be discarded. The experimental results show that the proposed model has an improvement in throughput up to 40%. Also, the CPU overhead and connection delay of the recommended technique is condensed by 20% when compared with an existing standard HCF and bloom filter. Future working in the field of the study comprises of addressing various forms of DDoS attacks such as web brute force attack, improvement in the multiple-class classification and self-configuration of the system, development of techniques for associating prompted alarms, and defensive measure formulation.

REFERENCES

[1] K. C. Okafor, J. A. Okoyeand G. Ononiwu, "Vulnerability bandwidth depletion attack on distributed cloud computing network: A QoS

RESEARCH ARTICLE

perspective". International Journal of Computer Applications, vol. 138, no. 7, pp.18-30, 2016.
[2] F. Shaarand A. Efe, "DDoS attacks and impacts on various cloud computing components", International Journal of Information Security Science, vol.7, no.1, pp.26-48, 2018.
[3] G. Somani, M. S. Gaur, D. Sanghi and M. Conti, "DDoS attacks in cloud computing: Collateral damage to non-targets", Computer Networks, vol. 109, pp.157-171, 2016.
[4] S. Yu, Y. Tian, S. Guo and D. O. Wu, "Can we beat DDoS attacks in clouds?", IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 9, pp. 2245-2254,2013.
[5] O. Terzo and L. Mossucca, "Cloud Computing with E-science Applications", CRC Press,2017
[6] A. M. Lonea, D. E. Popescu andH. Tianfield, "Detecting DDoS attacks in cloud computing environment", International Journal of Computers Communications & Control, vol. 8, no. 1, pp. 70-78, 2012.
[7] A. Bremler-Barr, E. Brosh and M. Sides, DDoS attack on cloud auto-scaling mechanisms", In IEEE INFOCOM Conference on Computer Communications, IEEE, pp. 1-9, 2017.
[8] P. Shamsolmoali and M. Zareapoor,"Statistical-based filtering system against DDOS attacks in cloud computing". In International Conference on Advances in Computing, Communications and Informatics, IEEE, pp. 1234-1239, 2014.
[9] J. Latanicki, P. Massonet, S. Naqvi, B. Rochwerger andM. Villari, "Scalable Cloud Defenses for Detection, Analysis and Mitigation of DDoS Attacks", In Future Internet Assembly, pp. 127-137, 2010.
[10] G. Somani, M. S. Gaur, D. Sanghi, M. Conti and R. Buyya, "DDoS attacks in cloud computing: Issues, taxonomy, and future directions". Computer Communications, vol. 107, pp. 30-48, 2017.
[11] A. R. Wani, Q. P. Rana, U. Saxena andN. Pandey, "Analysis and detection of DDoS attacks on cloud computing environment using machine learning techniques", In Amity International conference on artificial intelligence, IEEE, pp. 870-875, 2019.
[12] M. Darwish, A. Ouda L. F. Capretz, "Cloud-based DDoS attacks and defences", In International Conference on Information Society, IEEE, pp. 67-71, 2013.
[13] R. Saxena and S. Dey, "DDoS attack prevention using collaborative approach for cloud computing", Cluster Computing, pp. 1-16, 2019.
[14] V. Chouhan andS. K. Peddoju, "Packet monitoring approach to prevent DDoS attack in cloud computing", InternationalJournal of Computer Science and Electronic Engineering, vol. 1, no. 2, pp. 2315-4209, 2013.
[15] A. N. Jaber, M. F. Zolkipli, H. A. Shakir and M. R. Jassim, "Host based intrusion detection and prevention model against DDoS attack in cloud computing", In International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, Springer, Cham, pp. 241-252, 2017.
[16] H. Luo, Y. Lin, H. Zhang andM. Zukerman, "Preventing DDoS attacks by identifier/locator separation",IEEE Network, vol. 27, no. 6, pp. 60-65, 2013.
[17] R. Patgiri, S. Nayak andS. K. Borgohain, "Preventing ddos using bloom filter: A survey", arXiv preprint arXiv:1810.06689, 2018.
[18] N. Patani and R. Patel, "A mechanism for prevention of flooding based DDoS attack", International Journal of Computational Intelligence Research",vol. 13,no. 1, pp. 101-111, 2013.
[19] O. Osanaiye, H. Cai, K. K. R. Choo, A. Dehghantanha, Z. Xu and M. Dlodlo, "Ensemble-based multi-filter feature selection method for DDoS detection in cloud computing", EURASIP Journal on Wireless Communications and Networking, vol. 1, pp. 1-10, 2016.
[20] A. S. Navaz, V. Sangeetha andC. Prabhadevi, "Entropy based anomaly detection system to prevent DDoS attacks in cloud", arXiv preprint arXiv:2013,1308.6745, 2013.
[21] N. Jeyanthi, N. C. S. Iyengar, P. M. Kumar and A. Kannammal, "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, 2013.
[22] X. Wang, "Mitigation of DDoS Attacks through Pushback and Resource Regulation",In International Conference on Multimedia and Information Technology, IEEE,pp. 225-228, 2008.
[23] S. S. Chapade, K. U. Pandey and D. S. Bhade, "Securing cloud servers against flooding based DDoS attacks". In International Conference on Communication Systems and Network Technologies (CSNT), IEEE,pp.524-528, 2013.
[24] P. Shamsolmoali, M. A. Alam and R. Biswas, "C2DF: High rate DDOS filtering method in cloud computing", International Journal of Computer Network and Information Security, vol. 6, no. 9, p.43, 2014.
[25] A. Saravanan, S. Sathya Bama, S. Kadry and L. K. Ramasamy, "A new framework to alleviate DDoS vulnerabilities in cloud computing". International Journal of Electrical & Computer Engineering, vol. 9, no. 5, pp. 2088-8708, 2019.
[26] K. Shridhar and N. Gautam, "A prevention of DDoS attacks in cloud using honeypot". International Journalof Science and Research (IJSR), vol. 3, Issue. 11,pp. 2319-7064, 2012.
[27] K. Kalkan and F. Alagöz, "A distributed filtering mechanism against DDoS attacks: ScoreForCore", Computer Networks, vol. 108, pp.199-209, 2016.
[28] K. S. Bhosale, M. Nenova andG. Iliev, "The distributed denial of service attacks (DDoS) prevention mechanisms on application layer". In International Conference on Advanced Technologies, Systems and Services in Telecommunications (TELSIKS), IEEE, pp. 136-139, 2017.
[29] K. Srinivasan, A. Mubarakali, A. S. Alqahtani and A. D. Kumar, "A Survey on the Impact of DDoS Attacks in Cloud Computing: Prevention, Detection and Mitigation Techniques", In Intelligent Communication Technologies and Virtual Mobile Networks, Springer, Cham, pp. 252-270, 2019.
[30] G. Somani, A. Johri, M. Taneja, U. Pyne, M. S. Gaur and Sanghi, "DARAC: DDoS mitigation using DDoS aware resource allocation in the cloud",In International Conference on Information Systems Security, Springer, Cham, pp. 263-282, 2015.
[31] C. Jin, H. Wang and K. G. Shin, "Hop-count filtering: an effective defense against spoofed DDoS traffic", In Proceedings of the ACM conference on Computer and communications security,pp. 30-41, 2013.
[32] S. Lagishetty, P. Sabbu and K. Srinathan, "DMIPS-Defensive Mechanism against IP Spoofing", In Australasian Conference on Information Security and Privacy, Springer, Berlin, Heidelberg, pp. 276-291, 2011.
[33] S. S. Kolahi, A. A. Alghalbi, A. F. Alotaibi, S. S. Ahmed and D. Lad, "Performance comparison of defense mechanisms against TCP SYN flood DDoS attack", In Innovations in Computer Science and Engineering, Springer, Singapore, pp. 1-10, 2014.
[34] "Netwag Tool," 2007. Available: http://ntwag.sourceforge.net/.
[35] G. Dayanandam, T. V. Rao, D. B. Babu andS. N. Durga, "DDoS attacks—analysis and prevention. In Innovations in Computer Science and Engineering", Springer, Singapore, pp. 1-10, 2019.

Authors

Mr. A. Somasundaram is working as Assistant Professor, Department of Computer Science, Sree Saraswathi Thyagaraja College, India. He did his UG program; B.Sc., (Computer Science) in the same college, and completed his Masters's degree M.C.An at Dr.Mahalingam College of Engineering & Technology, Pollachi, India. He qualified himself in SET and NET examinations. Heis currently is pursuing a Ph.D. at Chikkanna Government Arts College affiliated to Bharathiyar University, Coimbatore. His research interests include Cloud Security and Distributed computing.

RESEARCH ARTICLE

Dr. V. S. Meenakshi is working as Assistant Professor at the Department of Computer Science, Chikkanna Government Arts College, Tiruppur, India. She is a recognized supervisor to guide Research Scholars of Computer Science at Bharathiar University. Presently she is guiding PhD scholars and produced 13M.Phil Scholars. She has published a good number of articles in national and international journals and acted as chairperson in national & international conferences. Her research area includes Biometric Template Security and Network Security.

How to cite this article:

A. Somasundaram, V. S. Meenakshi, "A Novel Three Layer Filtering (3L-F) Framework for Prevention of DDoS Attack in Cloud Environment", International Journal of Computer Networks and Applications (IJCNA), 8(4), PP: 334-345, 2021, DOI: 10.22247/ijcna/2021/209700.