Packet Resonance Strategy: A Spoof Attack Detection and Prevention Mechanism in Cloud Computing Environment

N. Jeyanthi[1], N.Ch.S.N. Iyengar[2]

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

Abstract: Distributed Denial of Service (DDoS) is a major threat to server availability. The attackers hide from view by impersonating their IP addresses as the legitimate users. This Spoofed IP helps the attacker to pass through the authentication phase and to launch the attack. Surviving spoof detection techniques could not resolve different styles of attacks. Packet Resonance Strategy (PRS) armed to detect various types of spoof attacks that destruct the server resources or data theft at Datacenter. PRS ensembles to any Cloud Service Provider (CSP) as they are exclusively responsible for any data leakage and sensitive information hack. PRS uses two-level detection scheme, allows the clients to access Datacenter only when they surpass initial authentication at both levels. PRS provides faster data transmission and time sensitiveness of cloud computing tasks to the authenticated clients. Experimental results proved that the proposed methodology is a better light-weight solution and deployable at server-end.

Keywords: DDoS, PRS, Cloud computing, Datacenter, Availability, Spoofing.

1. Introduction

Cloud computing supports resource abstraction i.e. the clients do not require any special hardware or software for complex operations. Cloud Data centers balance the load by supplying the necessary resources on-demand.

With continuous improvement in Cloud computing, security issues also grows along with it. Without any proper security solutions, the data that resides Datacenter may prone to attack by any assailer which ultimately results in data/ resource loss for the subscribers based on the service model preferred. As the precious cloud resources are handled by attackers' which results in loss of availability of Datacenter for clients and loss of revenue for cloud service provider.

Cloud Computing is a contour technology which combines several distributed networking technologies like distributed computing, virtualization and grid computing. So, this achieves the advantages of all the technologies on one hand. On the other hand, the security issues faced by these technologies will also affect this emerged technology. So, the design of any solution should adapt the cloud computing for earning the complete benefit with its additional characteristics like rapid elasticity.

Distributed Denial of Service (DDoS) is one of the serious security threats that challenge the availability of the Datacenter (DC) resources to the intended clients. The existing solutions are not that much effective to monitor the incoming traffic and to detect the DDoS attack if the attackers' traffic intensity is high. Therefore it is necessary to devise a mechanism for such situation to deactivate DDoS attackers in order to serve the legitimate users with DC resources. With DDoS attack, an attempt of identifying the source is almost impossible as several attackers tries to compromise the DC.

IP Spoofing so called *availability threat* is one such technique used by a hacker/ attacker to instigate DDoS attacks and gain control over server machine. Existing Spoof Detection mechanisms are *host-based* and *Router-based* whose performance depends upon the network behavior and attack strength [1].

Aim of this paper is to improve DC availability and allow DC to service only legitimate clients and to prevent other type of attackers' entry towards DC. This filtration achieves confidentiality, data theft prevention, DC resources protection which ultimately results in improved throughput with negligible delay in traffic analysis.

The proposed spoof attack detection algorithm focus on three different types of spoof attacks so called: Impersonation, Hiding attack, Reflection attack named as *Packet Resonance Strategy* (PRS). The traffic is generated by Email, HTTP, FTP applications. It has two levels namely *Packet Bouncer* and Packet *Transit* to identify the spoof attack threats. At level one, Packet Bouncer monitors the incoming traffic, applies detection mechanism, notifies and prevents swarm spoof attack. At level two, Packet Transit uses the packet information provided at level one and uses its own mechanism to notify and prevent the dwarf spoof attack. Each requester must successfully surpass this preliminary probing before accessing the DC resources.

Rest of the paper is organized as follows: Section 2 describes surviving techniques. Section 3 presents overview of the proposed architecture and methodology, Section 4 explains working mechanism Section 5 the performance of proposed mechanism, Section 6 the advantages of proposed approach and Section 7 conclusions with future work.

2. Related Work

Surviving spoof detection techniques motivated the development of PRS scheme. IP puzzle is the one of the method to mitigate the spoof attack. For every request to the server, the server sends the puzzle to the requester's source address. Spoof attacker will not receive the puzzle even if cannot solve to respond with any solution. The server establishes connection to the clients only when it receives the right solution [2]. TCP handshaking is one of the ways to detect spoof attack with a drawback that the sequence number could be easily cracked by the attacker.

Hop Count Filtering (HCF) [3], [5] explains the Filtering scheme where the hop counts are calculated for a trial. HCF maps the IP address to hop counts. If attacker spoofs, then this leads to hop count mismatch which is a significant characteristic. Due to the changing internet traffic, there can be change in legitimate clients' routing which creates hop count mismatch. Filtering due to this observation leads to false positives. So, the threshold is measured, the filtration is performed based on the threshold that minimizes false positives, but fails to detect the low rate attacks.

Dynamic key generation and incremental deployment makes this methodology a self-resilient against IP spoofing attacks. These features also make the system slower and vulnerable. Inter Domain Packet Filter depends on the shared BGP messages to validate the source address and protects the network from IP spoofing based DDoS attacks [4]. The root node which invoked the DDoS attack can be identified and blocked from spreading the attack with partial activity. Optimal routes are identified and shared with the neighboring nodes which can cause node-instability problem.

IP Spoofing attacks instigated at the access router level are detected and prevented in Trust-based Approach [6] using special centralized judge routers. Single point failure is the possibility here.

Route-based distributed packet filtering [7] prevents the spoofed IP packets to reach the destination. Route is used as parameter here which is not so reliable because route may change in real time. Scalability issue is also there as there is need of global knowledge of network infrastructure.

The Cloud Trace Back (CTB) [8] is a method where the detection is performed at the edge routers in between the clients and web servers. It marks the request from the client with CTB Marker within header. All service requests are first sent to CTB which prevents the direct attack on the web servers. When attack is detected the victim will ask for reconstruction to extract the mark. This will help in tracing the source. The cloud protector, the trained BPNN detects and filters the attack. However, the detection and filtering of attack starts only after the attack traffic reaches the victim.

Packet marking and altering in the Pi (Path Identifier) DDoS defense scheme are combined [9] to mark the packets on stack-based and write-ahead marking, replaced the holes with Pi-enabled routers in a path. Still it could not provide an error free solution.

Combined approach of the existing techniques resolved the deployment incentive problem of ingress filtering from a new, economic perspective [10].

The scheme proposed in [11] addresses the security issues related to data security by public key cryptography in Cloud computing. It also considers the issues like of data safety (service provider, internal users and from external attackers)

Fault tolerant work flow scheduling [12] makes use of failure probability information which tries DC to serve and available all the time. A checkpoint replication at each node rather than employing it in common node improves fault tolerance in cloud computing because the failure of central node will not crash the fault tolerant mechanism [13]. This tries to improve availability of DC. But with the growing number of attackers, the schemes poses a greater delay and lose the time-sensitiveness characteristic of cloud computing. Additionally, the scheme that supports cloud network should be scalable and should not create load at DC. Some of the schemes will work well for the distributed network but lags at cloud network which leads to Thrashing at high attack rate i.e., DC spends most of its time in detecting the traffic characteristic rather than servicing clients.

These drawbacks motivated us to propose an enhanced solution where DC employed with external hardware for improving detection rate and availability to legitimate clients.

3. PRS: Architecture and Methodology

This section describes our proposed architecture of PRS, general principles, and behavior of our proposed mechanism with different kind of incoming requesters.

3.1 PRS Architecture

Two levels of detection in Packet Resonance Strategy, PRS, are ***Packet Bouncer and Packet Transit***. At each level the attack traffic is detected, minimized and prevented at consecutive transmission through the DC channel. It can be presumed that the DC requesters are the combination of legitimate and spoof attackers. In the first level of PRS detection, Reflection Mirror Node (RMN) supports *Packet Bouncer* functionality, acts as a Reflective mirror where each incoming packet is logged and the packet of small size is bounced back to the requester with a random number. The requester should reply to it along with the same packet. This reply authenticates the requester, investigates the MAC and IP address combination and verifies the legitimacy of the requester. At this stage the reflection attack, hiding attack are detected and prevented, persisting traffic along with some packet information is passed onto next level, as shown in Figure 1.

In the second level of PRS detection, Transparent Mirror Node (TMN) supports *Packet Transit* functionality, acts as a transparent mirror. Here, the incoming requesters are inquired for an origin pass code that was created at the time of account creation. If the requesters fail in this validation, the impersonation spoof attack is detected and other legitimate traffic is allowed to access DC. So, the legitimacy is verified meticulously based on their behavior and authenticated with the origin characteristic.

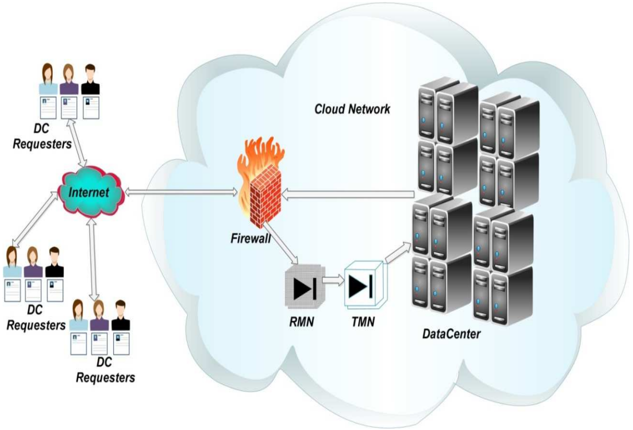

Figure 1. Architecture of the Proposed System

This two level filtering provides adequate amount of information to detect the legitimacy of any requesters and DC serves the legitimate clients faster in any further requests.

3.2 Rationale of PRS Architecture

DC Requesters can be a legitimate client or can be an attacker or can be a combined incoming traffic of legitimates and attackers. ***Firewall*** prevents the misbehaving requesters' entry into the server end. ***Reflection Mirror Node (RMN)*** continuously monitors the incoming traffic and validates the source address to detect swarm spoof attacks. ***Transparent Mirror Node (TMN)*** continuously monitors the traffic arriving from RMN and validates the source address to detect dwarf spoof attacks. ***Load Balancer*** configured to bypass only the compatible packets and the packets arriving to service the certain application (web page request, file download request, e-mail data download requests). ***Built-in Packet Analyzers of RMN and TMN*** extracts the header information and passes this information to firewall to prevent the packet entry from unauthorized requester until the session expires. The packet is now destroyed and further transmissions of packets are denied for the unauthorized requester. ***Data Centers*** are the Resource Provisioners, which always service only legitimate clients and even not for aggressive legitimate clients to improve availability.

When will PRS Allow Legitimate requesters: Legitimate requesters are clients who follow legitimate protocol pass the request packet probing at RMN, and with a validation of source address along with origin pass code at TMN. On successful authentication at RMN and TMN, the clients are considered to be legitimate.

When PRS Restrict Spoof attackers will: Attackers usually follows the legitimate protocol by learning the network behavior. But the intent of attackers is to launch uncontrollable spoofed packets and to shutdown the service of DC. So, the inter-arrival pattern, back-off timer expiry, number of attempts failed and responsive attempts reveals the attackers characteristic. Hence, any misbehaving requesters are strictly restricted. This validation is adequate to decide the requesters as an attacker, as the packet probe completely validate the incoming requester. By logging and comparing the incoming requesters' packet and the bouncing packet with random number reveals the swarm spoofing attacks (Hiding, Reflection) and dwarf spoofing attacks (Impersonation).

When to Accept the New client's requisition: Supremacy in PRS detection scheme is that, the new requesters are treated as attacker until they are successfully validated at RMN and TMN. This leads us to improve the detection accuracy in our approach. Once the new requester becomes a registered legitimate requester and follows legitimate profile, higher priority is assigned in session table to service them quicker rather to unregistered client requesters.

4. Working Mechanism

This section describes the flow of PRS working mechanism, different attack types considered in our proposed mechanism, and the modular description of schematic detection and prevention mechanism.

4.1 Flow Diagram of PRS Algorithm

DC requesters are allowed to enter into cloud network only through the firewall. Requesters will be blocked at the firewall when there is discrepancy in the MAC and IP address combination.

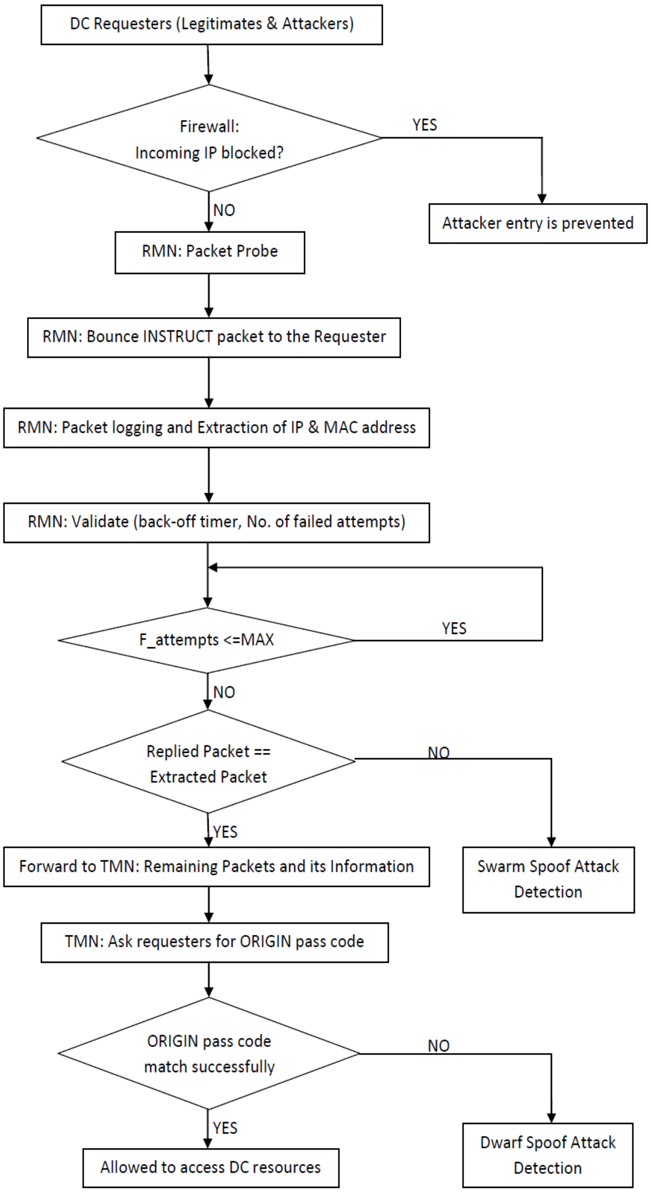
DC Requesters (Legitimates & Attackers)
Firewall: Incoming IP blocked?
YES
Attacker entry is prevented
NO
RMN: Packet Probe
RMN: Bounce INSTRUCT packet to the Requester
RMN: Packet logging and Extraction of IP & MAC address
RMN: Validate (back-off timer, No. of failed attempts)
F_attempts <=MAX
YES
NO
Replied Packet == Extracted Packet
NO
YES
Forward to TMN: Remaining Packets and its Information
Swarm Spoof Attack Detection
TMN: Ask requesters for ORIGIN pass code
ORIGIN pass code match successfully
NO
YES
Allowed to access DC resources
Dwarf Spoof Attack Detection

Figure 2: Working mechanism of PRS Algorithm

If the DC requester does not perform any attack activity, they are allowed to enter in at RMN. Otherwise, the packet probe begins by logging the packet information and INSTRUCT packet is bounced which is a special packet to validate the requester legitimacy. Based on the response, the swarm spoof attack is detected. Remaining validated packets are forwarded to TMN, so that the Impersonation attack is monitored by sending a sealed sequence number and asking for ORIGIN pass code as shown in figure 2. This response helps in detecting the man-in-the-middle attack. On validating the requester as a legitimate client, they are serviced through a secured channel. This channel is free of attack. RMN and TMN are connected to DC, the detection mechanism acts as intermediary approach rather than end-host based approach.

4.2 Types of Spoof attacks

Among the several types of spoofing attacks, the following attacks are addressed as they are launched on behalf of clients and destruct the DC resources.

Type I: Hiding attack

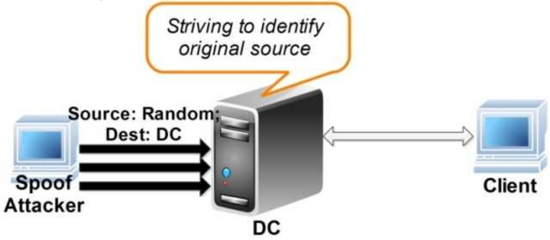

Figure 3. Hiding attack

In figure 3, Attackers simultaneously send a large number of spoofed packets with random IP address. This creates chaos at DC to process which specific packets as legitimate packets.

Type II: Reflection attack

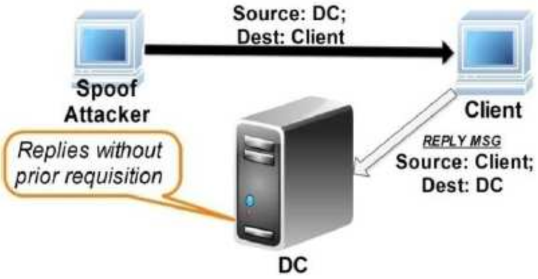

Figure 4. Reflection Attack

In figure 4, the attackers send spoof packets with source IP address as victim to any unknown user. This causes unwanted responses reaching the victim from unknown user and increases the flood rate.

Type III: Impersonation attack

In figure 5, the attackers send spoof packets with the source IP address of any unknown legitimate user and acts as a legitimate user. This is equivalent to man-in-the-middle attack. Spoof attacker receives requests from client and spoofs IP and forwards the requests to DC acting as legitimate user.

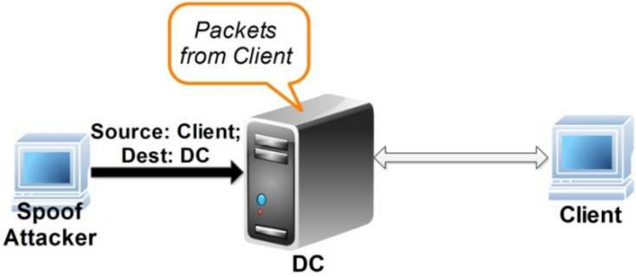

Figure 5. Impersonation Attack

The responses of DC are again processed intermediately and send to clients. This leads to confidentiality issues and data theft / data loss at DC.

Black arrows in figure 3, 4, 5 represent the spoof packets. If no proper detection mechanism is not in place, the DC could respond badly or lead to partial shutdown of services.

4.3 Design of Spoof Detection and Prevention Algorithm in PRS

DDoS, a serious security threat and hard to detect as it involves several distributed attackers. It is feasible to detect and prevent such threats than identifying attack source launchers. The several ways to launch this type of attack are:

- *Create a Botnet to launch attack against the attack target*
- *Hundreds of distributed human attackers*
- *Spoof the existing legitimate clients*

Of all these kind of attacks, spoofing is the only type where the methodologies like tracking the source is almost impossible as the attacker disguise as another legitimate.

The proposed detection algorithm would detect the attackers as early as possible and outwit them from further accessing of DC resources as explained in this Section.

4.3.1Traffic Monitoring

Whenever the DC requesters direct the request packets, they have to be monitored and classified as normal or abnormal traffic, prior to detection. Based on the incoming traffic, the decision could be made whether the incoming traffic is normal or abnormal. If the traffic condition is normal, they are forwarded without detection, this leads (type III Impersonation attack) to reach DC. Only when there is an abnormality in the traffic condition, they are forwarded to a separate module for detection. But this issue is eliminated in our scheme because RMN acts as spoof anomaly detector and also as traffic analyzer. So, RMN keeps track of packet characteristics and their pattern for each client which lets RMN to detect the behavior of each requester.

4.3.2Swarm spoof attack detection

Swarm spoof attacks are of two types such as Type I Hiding and type II Reflection This attack detection is performed at RMN (Level 1 detection).

Algorithm 1: Swarm spoof Attack Algorithm

```
Input: Incoming traffic packet
Output: Type I and Type II attack detection.
begin
    Foreach (incoming packet)
            Packet_Bounce ();
            PacketInfo_Log ();
            Packet_Extraction ();
            Src_addr_Validation ();
end
```

Every incoming requester’s packet that reaches RMN will be probed by the ***Packet_Bounce phase*** of Algorithm 1. Reflection Mirror node, RMN, bounces the INSTRUCT packet (which is very small size and has random number usually 16 to 64 bit and is valid only for that particular bounce) to the requester immediately and waits for its reply over a short period of time (usually time-out period). This can be achieved by triggering the back-off timer. The random number of this small length can be very well relied as we maintain back-off timer for each bounced packet. The probability of cracking the random number is very less because once the timer expires, the INSTRUCT packet also expires. At the same time, increase in back-off timer period could cause overhead to the other buffered requesters and imposes huge delay. To avoid buffer queue overhead and delay, timer value is maintained as small as possible.

On bouncing the packet, RMN logs the packet information at back-end in the ***PacketInfo_Log phase***. RMN logs the details (tabulated below in Table.1) of the incoming traffic that helps in identifying the requester behavior.

Table 1. Details of RMN Log fields and their purposes

RMN Log field	**Purpose**
Client IP	For Later Verification
Inter-arrival time of each packet	Discriminates the incoming client as legitimate or triggers flood
Request packet type and size	Aids RMN traffic analyzer to monitor the compatibility with application specific requests
Bounced packet	Contains the information about the generated random number and the requester whose INSTRUCT packet has been bounced
Back-off timer	Notifies the time-out for each requester
Number of Failure attempts	Distinguishes attacker and legitimate

The limit (number of packet logging per second) of the packetInfo_Log can be set to a larger number at high attack prone zones which in turn improves the processing capability and attack detection proportionately.

On successful logging, the packet is extracted for identifying MAC address in the ***Packet_Extraction phase***. MAC address extraction performed at the back-end helps in detecting the attacker appropriately with a negligible amount of delay. The attacker with spoofed IP easily intrudes the cloud and triggers flooding at a high rate. Hence, it is mandatory to have an unambiguous identification for each client. MAC address and IP address combination of each client could serve the purpose. In real-time, we suggest considering the browser session ID to MAC and IP address combination of each machine because the same physical machine could work with virtual machines or the same client can also log in to two different browsers of same machine.

Src_addr_Validation is an important phase in detecting the spoof attackers. The response received for the bounced INSTRUCT packet is matched with the RMN triggered random number. Matching could partly validate the source, but is not a perfect detection. There is a possibility for failure if the spoofed attacker replies with random number, though it is very rare. The INSTRUCT response packet's MAC and IP address combination is compared to the requesters' initial request's MAC and IP address. On successful match, they are validated at RMN (level 1) detection and are forwarded to TMN (level 2) detection.

At this level, the Hiding spoof attacker and Reflection spoof attacker is detected and dropped. The IP and MAC combination information is sent to firewall for preventing their further entry. It is advisable that for any high attack prone zone, the back-off timer, number of failure attempts could be made less for improving the availability of DC. Otherwise, attack requesters gain advantage with prolonged Time-out and number of failure attempts. At this stage, all the Hiding and Reflection kind of spoof attack is detected. Combination of MAC and IP address with INSTRUCT packet validation is unique for each requester at any time and this assures the type I and II attack detection. Remaining traffic is fed to TMN for further processing. This level of detection identifies the spoof attack that flood at high rate. Now, the next level of detection at TMN processes the remaining traffic for other attack detection.

4.3.3Dwarf spoof attack detection

Dwarf spoof attacks are type III attack (Impersonation). Combination of MAC and IP of any incoming requesters' is only a part of our detection to outwit swarm spoof attack type. Rather there are attackers' who learns network behavior and launch spoof attack which resembles within legitimate profile. They are the attack source which floods the attack target with arbitrary attack packets and follows legitimate profile which cause low rate flooding attack. These are weak attacks result in client's data theft and other confidentiality related issues. This attack detection is performed at TMN (Level 2 detection). Even the intelligent spoof attack activity is detected at this stage with the help of sealed sequence number and origin pass code validation.

Algorithm 2: Dwarf spoof Attack Algorithm

```
    Input: Remaining traffic packets and its information
           after Type I and Type II attack detection.
    Output: Type III attack detection and legitimate client
           classification
begin
    Foreach (incoming packet)
        Packet_process ();
        Origin_passcode_quest ();
        Passcode_validation ();
        Packet _Transit ();
end
```

Table 2. Details of TMN Log fields and their purposes

TMN Log field	**Purpose**
RMN Flag	• Validates the packet information that arrives from RMN. • Set to 1 for successful logged packets
Sealed sequence Number	• Necessary for retrieval of requester origin pass code • Improves confidentiality • Contains the sequence number • Tracks number of times the packet is processed • Initially set to 0. Tracker incremented by 1on each packet extraction • TMN should receive tracker ID with 1 to obtain confidentiality. Otherwise, Alternate channel can be chosen for further communication • Also detects the man-in-the-middle attack.
Origin pass code response	Secured code, created on account creation and not known to any other individuals.
Number of Pass code attempts	incremented until the maximum number of attempts reached

On successful logging of packet information, TMN's **Origin_passcode_quest** phase in Algorithm 2 quests the origin pass code for authenticating the legitimate clients. Pass code is created only for the legitimate clients at the time of account creation, acts as additional security code. This origin pass code can have some protocol to be stronger. It is write-protected even for the account holder to improve the detection accuracy. Write-protected pass code could easily predict the legitimate client and any attempt to change in origin pass code or incorrect pass code could be considered as spoof attack. We suggest the usage of some light-weight crypt key for transmission of origin pass code.

The requesters must respond with the acknowledgement of sealed sequence number and also the origin pass code in ***Passcode_validation***. The received pass code has to be compared with database and validated, if the validation is successful, the clients are considered to be legitimate and spoof attack otherwise. The special case is obtaining the right acknowledgement for the sealed sequence number but with incorrect origin pass code. At this juncture, we must monitor the inter-arrival time of such requester. If inter-arrival time is very less (imitates DDoS attack), the packet is dropped. Otherwise, the client can be given some limited number of attempts and if the maximum number of attempts reached with incorrect pass code, the requester is considered to be a spoof attacker (Impersonation attack) who follows legitimate protocol.

On validating the requester's packets, it is feasible to allow the packet to access DC in ***Packet _Transit phase***. If any deviation in inter-arrival time is observed, they are dropped.

In this stage, TMN detects the Impersonation spoof attack based on the pre-extracted packet's information, sealed sequence number, origin pass code validation. This helps in detecting the Impersonation spoof attack and man-in-the-middle attack. Thus at TMN (level 2) detects dwarf spoof attack i.e., attack threat but with less strength usually launched for any data theft at DC end or the clients' sensitive data. The authentication marked based on MAC and IP combination helps in rapid transmission for legitimates until the legitimate session expires.

4.3.4Spoof attack prevention

As is it tough to detect the attacker, they are to be outwitted immediately to prevent any serious disaster at DC end. So, we employ firewall which continuously monitors the incoming traffic. When any abnormalities found, they are prevented by restricting their entry and dropping their packets at firewall. This in turn improves the availability of DC only to legitimate clients.

Algorithm 3: Spoof Attack Prevention

```
Input: Incoming traffic packets
Output:  Attacker entry prevention.
begin
    Foreach (incoming packet)
            Packet _source_addr_validation ();
            Decision ();
end
```

As we deal with spoof detection algorithm, we can't rely on IP address, so, we add a new functionality of extraction and validation in Algorithm 3 named ***Packet _source_addr_validation***. Based on the packet MAC and IP address we detect the spoof attack threat. So, on threat detection, the MAC and IP combination is forwarded to firewall, further flooding towards DC from any such threat initiators are restricted based on source address validation.

In the ***Decision phase*** of Algorithm 3**,** if the attack sources' MAC found in firewall Access_Control_Restriction list, then the attacker is prevented from accessing the DC resources and the packets are dropped off. For efficient searching, the addresses could be placed at hash tables. Firewall also acts as a preliminary traffic analyzer to prevent the attacker MAC address source until their session expiry.

Prevention is the mechanism where the identified attackers are immediately restricted at DC end. The important MAC and IP address combination reveal the attacker characteristic. The attackers are blocked until their session expiry. The packets are blocked based on the MAC address at firewall. The requester could only reach RMN after the current session expiration.

Our algorithm detects the various spoof attack threat at various levels systematically. Firstly, it detects the attackers who launch the attack at high rate and the traffic is considerably reduced at next level to detect the low rate attack. Once this initial authentication scheme is validated, the attackers are detected and outwitted. Successive legitimates transmissions will have no delay and directed towards DC, where DC process only legitimate client requests. This way the proposed Packet Resonance Strategy (PRS) Algorithm detects and prevents the attackers' further entry by monitoring at the firewall.

5. Experimentation and Performance Evaluation

This section describes the experimental scenario and performance analysis with important factors that highlights the advantages and necessity of the PRS deployment for detecting and outwitting the spoof threats at DC end.

5.1 Experimental setup

We tested our proposed mechanism as simulation experiment in OPNET Modeler as per [14], [15], [16]. The experiments are performed in a campus network where DC requesters are grouped in three subnets and each subnet has got 400 workstations. 400 attackers and 1000 legitimate clients requesting for application-specific requests at each subnet. This way we created the attacker and legitimate profile and other devices which would be needed to test our algorithm as an experiment. The traffic represents internet and the group of spoof attackers are activated at varying time intervals. The attack profile is replicated to increase the attack strength to engage the DC resources like bandwidth, CPU, Memory. On the whole, our experiment has 3000 clients and 1200 attackers. But we also evaluated with different number of attackers to measure the detection strength.

5.2 Performance Evaluation

5.2.1 Email Response Time

Email response time is the statistic measured as the time elapsed between sending requests for emails and receiving emails from email server in the network. This time includes signaling delay for the connection setup.

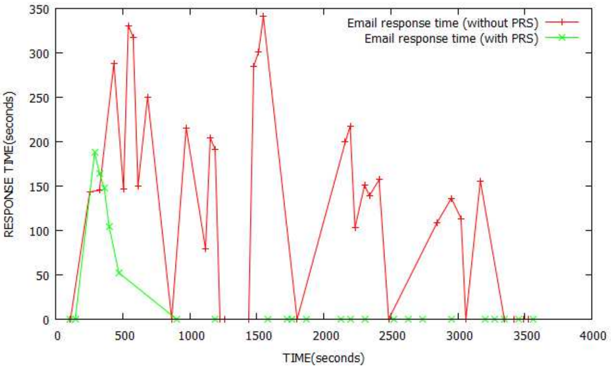

Figure 6. Email Response time

Figure 6 shows the response time of legitimate email requests of DC without PRS and DC with PRS. Increase in application-specific (Email, FTP, HTTP) response time symbolizes the poor performance of DC. DC without PRS shows frequent spikes in responding email request of size (100 KB – 2 MB) can be seen in figure 6 shows DC serves the intended clients very poorly. In contrast to former, DC with PRS initially took time in detecting the attacker behavior which can be seen as a spike and it inclines gradually, due to periodical activation of other attackers. On successful detection, they are outwitted by blocking at firewall which results in quicker response to its intended clients.

5.2.2 FTP Response Time

FTP Response time is the statistic measures as the time elapsed between sending a request and receiving the response packet. Measured from the time a client application sends a request to the server to the time it receives a response packet. Every response packet sent from a server to an FTP application is included in this statistic.

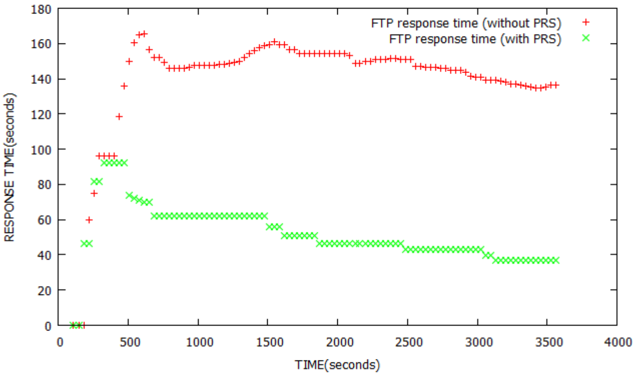

Figure 7. FTP Response time

Figure 7 shows the response time for FTP data download request of DC without PRS and DC with PRS. DC without PRS creates a steep raise in response time for FTP requests of size (500 KB – 5 MB) and it never fall down because of the attackers' spoof launch which can be seen in figure 7. In contrast to the former, DC with PRS steeps and tries to reduce response time by outwitting attackers at each time interval which gradually provides better response time to requesters because of varied size of data downloads requests.

5.2.3 HTTP Response Time

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

Figure 8 shows the response time of HTTP requests of DC without PRS and DC with PRS which are usually in size of about 10 KB- 200 KB. DC without PRS shows the similar attack pattern that is equivalent to email attack pattern in figure 6. As the request size is comparatively less for HTTP, so is the response time of application. In contrast, DC with PRS still responds better to the HTTP requests from the origin than DC without PRS in figure 8.

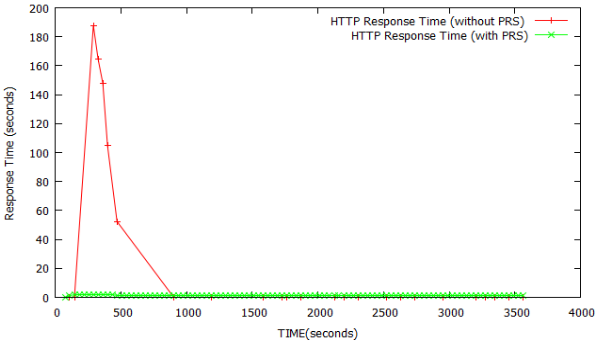

Figure 8. HTTP Response Time

5.2.4 Task Load at DC

Task Load represents the current number of Application sessions on the DC. This statistic is intended to provide you a picture of how loaded the server is with Application sessions. Here Tasks/sec actually correlates to Sessions/sec.

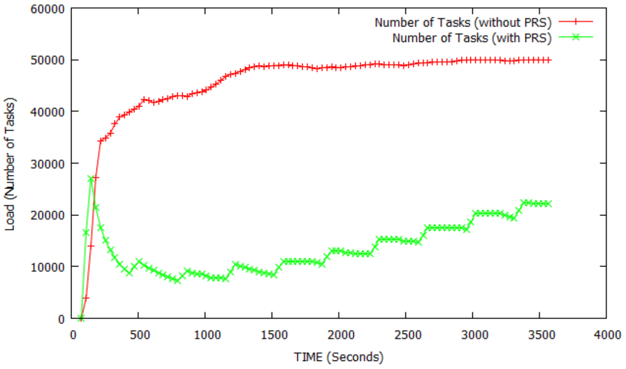

Figure 9. Task Load at DC

Figure 9 shows the load in terms of application specific tasks. DC without PRS imposes huge number of load at DC as the incoming traffic is a combination of legitimate and spoof attacker. In contrast to former, DC with PRS creates sessions only for legitimates where the session update (periodical time out) is carried out at regular intervals even on increase in number legitimates and their task load. The gradual oscillation shows the session creation and expiry at DC which results in normal behavior of DC, which is shown in figure 9.

5.2.5 Link Throughput (bps)

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

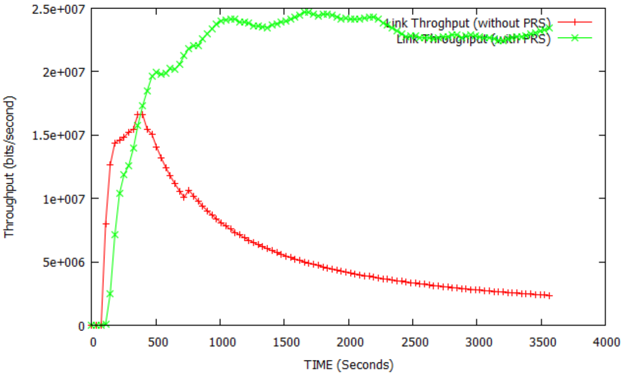

Figure 10. Link Throughput (bps)

Figure 10 shows the link Throughput of DC without PRS and DC with PRS in terms of bps (bits per second). DC without PRS could not detect the attackers' traffic flood which leads to reducion in legitimate traffic throughput as the link is completely flooded with distributed spoof attacker. In contrast to former, DC with PRS detects the attackers' behavior and eliminates them which in turn allow legitimate traffic to reach the DC without any rigorous processing overhead.

5.2.6 Link Throughput (pps)

Link Throughput is the statistic that represents the average number of packets successfully received or transmitted by the receiver or transmitter channel per second. As the traffic includes both legitimate and attack pattern we consider only the legitimate traffic packets that reaches the DC and recorded at each transaction.

Figure 11 shows the link Throughput of DC without PRS and DC with PRS in terms of pps (packets per second). Figure 10 and figure 11 has the same pattern as they represent the average legitimate traffic throughput but with different units. The intention of figure 11 is to prove that the packet processing per second of legitimate packets at DC with PRS is quicker than DC without PRS

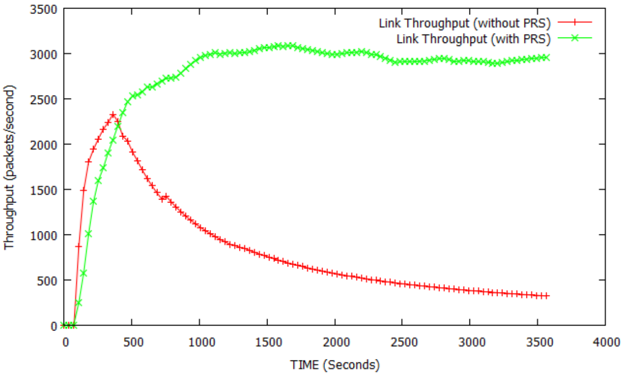

Figure 11. Link Throughput (pps)

5.2.7 CPU Utilization

CPU Utilization is the statistic reports the utilization, in percentage (%), of the 'CPU'. This statistic measures the utilization of central CPU only. It does not measure the utilization of CPUs used for IP slot processing.

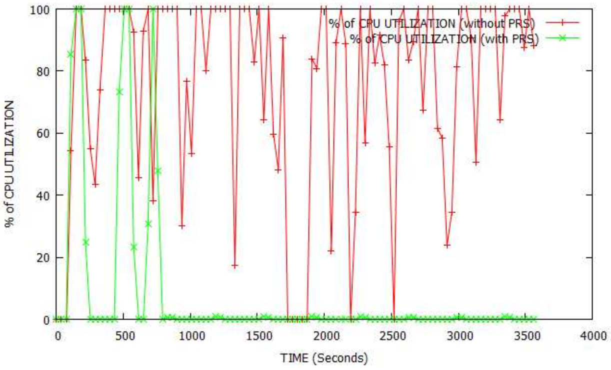

Figure 12. CPU Utilization

Figure 12 shows CPU utilization of DC without PRS and DC with PRS. CPU utilization is the rate of CPU usage. So, CPU utilization discussed here really means the stress at DC, as each DC employed with several physical hosts. DC without PRS shows vigorous oscillation of CPU usage rate represents partial shutdown of service or poor service to its intended clients which can be noticed as 100% CPU usage rate in figure 12. In contrast to former, DC with PRS imitates attack pattern for short period of time, after the detection of spoof attackers, the utilization reaches normal level and continues to serve legitimates. Small peaks after detection, shows the application-specific tasks completion and acquiring other tasks from other requester. This task completion indicates the session expiration. This task completion time is less for legitimates and more for attackers because of attackers' request rate and size.

5.2.8 Active legitimate connections

Active legitimate connections are the statistic which is measured as the total number of active legitimate clients connected to the DC that are logged at each point of time. Figure 13 shows the number of legitimate clients connected to DC with PRS and DC without PRS.

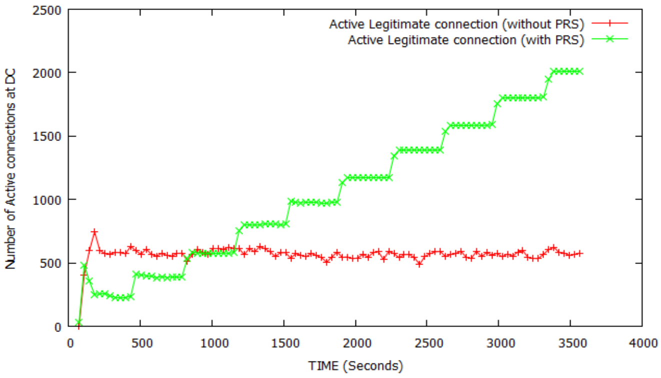

Figure 13. Active Legitimate Connections

DC without PRS shows the numbers of active legitimate clients' connections are restricted to very less connections which symbolize the poor response and increased delay to the legitimate clients. In contrast, DC with PRS constantly acquires the number of legitimates connection at DC and exponentially increase because of the attacker entry restriction. Figure 13 shows the performance have been improved by logging more number of legitimate connection at DC with PRS which ultimately shows the performance is 3 times better than the DC without PRS at the time of spoof attacks.

5.2.9 Retransmission count

Retransmission count is the statistic that measures the total number of TCP retransmissions in the network. Written when data is retransmitted from the TCP unacknowledged buffer. This statistics indirectly shows the failure rate of legitimate connection.

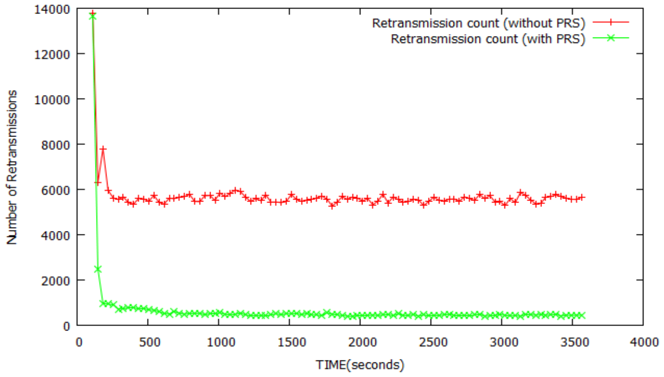

Figure 14. Retransmission Count (Failure Rate)

Retransmission is a symbol of failure of response from the requester or the sent request is broken, this characterization is shown in figure 14. Increase in retransmission results in delay in connection and resources (Time and Memory) wastage. Attackers populate requests towards DC and if DC does not receive response, retransmission is done. This indirectly creates a denial of service for other waiting requesters. At figure 14 DC without PRS have more retransmissions indicates huge loss of packet delivery and results in denial of service for buffered requesters. Contrast to former, DC with PRS have less retransmissions and improves the packet delivery fraction. This proves the number failures are more at DC without PRS and DC with PRS works well as the number retransmissions are restricted (based on the MAC and IP, ORIGIN pass code) and further requisition is blocked at firewall which proves the efficiency in detection.

5.2.10 Number of connections Aborted

Number of connections aborted is the statistic that measures the total number of legitimate connections aborted by huge traffic. Increasing the number of connections at each time a TCP connection is aborted at this node.

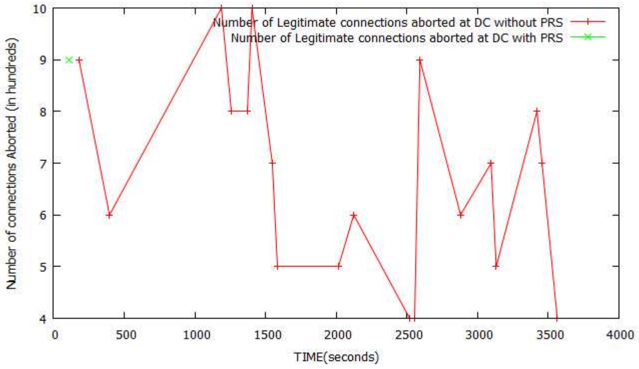

Figure 15. Number of Aborted Connections

Figure 15 shows the number connections aborted at DC. Connection abortion is the result of resource unavailability which is the characteristic of DDoS attack. Launching distributed spoof attack is also a characteristic of DDoS attack. DC resource without PRS is continuously destructed by attackers' request and leads to connection abortion at each of time. Whereas at DC with PRS, the detection based on the behavior took small fraction of time, and later the connections are not aborted. This shows DC employed with PRS has no symbol of DDoS after the attacker detection. Though the attackers are activated dynamically, their MAC and IP combination reveals the attack characteristic, so their requests are dropped until their change in behavior. DC with PRS destroyed 900 connections initially because they require some behavior of requester to detect. DC without PRS, the connections are aborted endlessly because of swarm spoof attackers.

Overall performance evaluation shows the better results for application-specific response time, reduced task load, increased the legitimate connections to DC and considerable reduction in legitimate connection abortion. Retransmission count reduction proves less failure for any legitimate connections.

6. Advantages of Proposed Model

We have discussed sufficiently regarding the performance that we evaluated in our experiment. One of the most important advantages is the ability to detect earlier based on the behavior. We have proved the response time and other important attributes are efficient than listed out in [2]. We have also experimented with the varied number of attackers and found the detection strength is suitable for improving the Quality of Service in cloud computing. Detection Strength is the statistic which measures the number of active attackers at any of time.

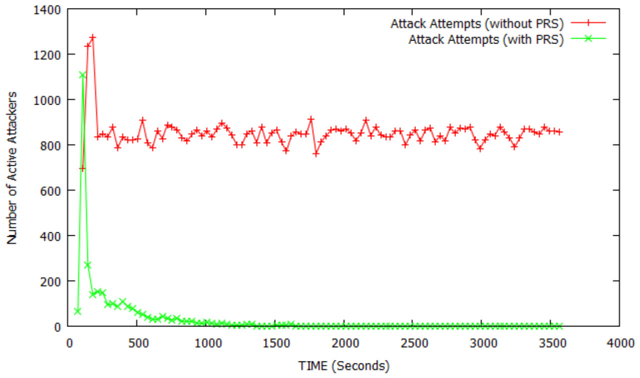

Figure 16. Detection Strength

Attack attempts are high initially and these attempts were blocked at firewall at later sessions. This proves our mechanism works better even with 1200 distributed spoof attackers. The intended RMN and TMN not only detect attackers based on the packet bounce. They also log the details like packet inter-arrival time which when combined with MAC address resolute the attack scenario and the attacker is detected. After validating at RMN, the attack threats are neglected and less number of packets is forwarded to TMN, as the number of attackers is outwitted at earlier level of detection.

In addition to the improved detection strength, we have also identified other benefits that would improve the choice of deployment.

- ***Delegated Detection Deployment* –** Special nodes namely RMN, TMN were deployed whose work is only detecting the incoming threats**.** This delegate approach avoids thrashing and improves detection strength.
- ***Hierarchical filtration* -** Incoming traffic is logged to RMN and packet probing begins at this phase. On detecting the swarm spoof attack they are prevented by dropping. At next level, valid packets are logged and examined by updating with TMN's requirements. Here, the dwarf spoof attack is detected and prevented. Thus at each level of filtration, some attack targets are deactivated.
- ***Improved confidentiality* -** The use of sealed sequence number and ORIGIN pass code and its write-protected policy allows satisfying the confidentiality. So, the security code is always confidential. Intentional packet segregation/ man-in-the-middle attack ia also easily detected.
- ***Improved Availability* -** Ability to detect the incoming attackers, validating them thoroughly and servicing the legitimate clients in secure channels enables resource availability and serviceability to all the requesters who bypass the legitimacy validation at RMN and TMN.
- ***Reduced Traffic congestion* -** Detecting and outwitting attackers at earlier time paves congestion free network for the legitimate clients.
- ***Better resource protection* -** As we are able to restrict the attacker before accessing DC resources, the resources like CPU, VM, RAM, storage can be protected from attackers and supplied to intended clients who indirectly improves fame of CSP and directly saves revenue cost.

7. Conclusion and Future Work

Spoofing attacks are still prevailing attacks as a kind of Denial-of-Service. There are several solutions available which has their own advantages and disadvantages. We have proven that our solution works well for all cases we considered. The proposed solution works well against any session hijack, random flooding and hidden massive flooding. Instead of authentication and creating the state for every incoming packet, they are examined at their initial stage. In order to prevent overhead, we deployed two separate nodes which employs the detection mechanism quicker. Our two levels of filtration detects and mitigates the traffic passing from one level to other level and at each level some types of attacks are detected and they are prevented precisely by the MAC and IP address combination, origin pass code authentication with sealed sequence number. The remaining traffic alone is passed onto next level. This strategy achieves quicker mitigation.

We have developed here a host-based detection mechanism, which is easier to deploy. Our future work is to enhance and adopt it at router level to detect the attackers before they reach the end host by considering all types of amplification spoof attacks like client-end, DNS amplification attack.

References

[1] Toby Ehrenkranz, Jun Li, "On the state of IP spoofing defense", ACM Transactions on Internet Technology, Vol. 9, No. 2, pp. 1-29, 2009.

[2] Chang Feng, W., Kaiser, E. C., Chi Feng, W., and Luu, A., "Design and implementation of network puzzles", In Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies, pp. 2372–2382, 2005.

[3] Jin, C., Wang, H., and Shin, K. G., "Hop-count filtering: An effective defense against spoofed DDoS traffic". In Proceedings of the Conference on Computer and Communications Security, pp. 30–41, 2003.

[4] Zhenhai Duan, Xin Yuan, Jaideep Chandrashekar, "Controlling IP Spoofing through Interdomain Packet Filters", IEEE transactions on Dependable and Secure computing, Vol. 5, No.1, pp.22-27, 2008.

[5] Haining Wang, Cheng Jin, Kang G. Shin, "Defense Against Spoofed IP Traffic Using Hop-Count Filtering", IEEE/ACM Transactions on Networking, Vol. 15, No. 1, pp.40-57, 2007.

[6] Jesus M. Gonzalez, Mohd Anwar, James B.D. Joshi, "A Trust-based Approach against IP-spoofing Attacks", IEEE Ninth Annual International Conference on Privacy, Security and Trust, Montreal, QC, pp.63-70, July 2011.

[7] Kihong Park, Heejo Lee, "On the effectiveness of route-based packet filtering for distributed DoS attack prevention in power-law internets", ACM Proceedings of the 2001 Conference on Applications, Technologies, Architectures and Protocols for Computer Communications, Ontario, Canada, Vol. 31, No.4, pp.15-26, 2001.

[8] Noureldien A. Noureldien Mashair O. Hussein, "Block Spoofed Packets at Source (BSPS): A method for Detecting and Preventing All Types of Spoofed Source IP Packets and SYN Flooding Packets at Source: A Theoretical Framework" IEEE Second International Conference on the Applications of Digital Information and Web Technologies, London, pp.579-583, August 2009.

[9] Abraham Yaar, Adrian Perrig, Dawn Song, "StackPi: New Packet Marking and Filtering Mechanisms for DDoS and IP Spoofing Defense", IEEE Journal on Selected Areas in Communications, Vol.24, No.10, pp. 1853 – 1863, 2006.

[10] Bingyang Liu, Jun Bi, Xiaowei Yang, "FaaS: Filtering IP Spoofing Traffic as a Service", ACM SIGCOMM'12, Helsinki, Finland, Vol.42, No.4, pp.113-114, August 2012.

[11] Karamjit Singh; Isha Kharbanda; Navdeep Kaur, "Security issues occur in Cloud Computing and there Solutions", International Journal on Computer Science and Engineering, Vol. 4, No.5, pp. 945-949, 2012.

[12] Jayadivya S K; Jaya Nirmala S; Mary Saira Bhanu S, "Fault tolerant workflow scheduling based on replication and resubmission of tasks in Cloud Computing", International Journal on Computer Science and Engineering, Vol. 4, No.6, pp. 996-1006, 2012.

[13] Sanjay Bansal, Sanjeev Sharma and Ishita Trivedi, Mrinalika Ghosh, ” Improved Self Fused Check pointing Replication for Handling Multiple Faults in Cloud Computing”, International Journal on Computer Science and Engineering, Vol. 4, No.6, pp. 1146-1152, 2012.

[14] Qwasmi, N.; Ahmed, F.; Liscano, R., “Simulation of DDoS Attacks On P2P Networks”, IEEE 13th International Conference on High Performance Computing and Communications, Banff, AB, pp. 610-614, 2011.

[15] Jha, R.K; Dalal, U.D, “On demand cloud computing performance analysis with low cost for QoS application”, International Conference on Multimedia, Signal Processing and Communication Technologies, Aligarh, pp. 268-271, 2011.

[16] Rakesh Kumar Jha, Upena D Dalal, “A performance Comparison with cost for QoS Application in On-Demand Cloud Computing”, IEEE Recent Advances in Intelligent Computational Systems, Trivandrum, pp. 11-18, 2011.