A pioneer scheme in the detection and defense of DrDoS attack involving spoofed flooding packets

Kavisankar L[1] , Chellappan C[1], Sivasankar P[2], Ashwin Karthi[1], Srinivas A[3]
[1] Department of Computer science and Engineering, Anna University
Chennai 600025 – India
[2]Electronics Engineering Department, NITTTR
Chennai 600113 – India
[3]Department of Information Technology, Madras Institute of Technology, Anna University
Chennai 600044 – India
[e-mail: kavisankaar@gmail.com]
*Corresponding author: Kavisankar L[1]

Received December 15, 2013; revised March 19, 2014; accepted April 12, 2014; published May 29, 2014

Abstract

DDoS (Distributed Denial of Service) has been a continuous threat to the cyber world with the growth in cyber technology. This technical evolution has given rise to a number of ultra-sophisticated ways for the attackers to perform their DDoS attack. In general, the attackers who generate the denial of service, use the vulnerabilities of the TCP. Some of the vulnerabilities like SYN (synchronization) flooding, and IP spoofing are used by the attacker to create these Distributed Reflected Denial of Service (DrDoS) attacks. An attacker, with the assistance of IP spoofing creates a number of attack packets, which reflects the flooded packets to an attacker's intended victim system, known as the primary target. The proposed scheme, Efficient Spoofed Flooding Defense (ESFD) provides two level checks which, consist of probing and non-repudiation, before allocating a service to the clients. The probing is used to determine the availability of the requested client. Non-repudiation is taken care of by the timestamp enabled in the packet, which is our major contribution. The real time experimental results showed the efficiency of our proposed ESFD scheme, by increasing the performance of the CPU up to 40%, the memory up to 52% and the network bandwidth up to 67%. This proves the fact that the proposed ESFD scheme is fast and efficient, negating the impact on the network, victim and primary target.

Keywords: Amplification attack, DDoS, DrDoS, IP Spoofing, SYN spoofing

The experimental work was carried out in the SSE Lab, Anna University Chennai, established with the funding supported by the NTRO, Government of India, for a collaborative project on "Collaborative Directed Basic Research on Smart and Secure Environment", and this paper has been prepared with the resources of this project. The authors would like to thank the project coordinators and the NTRO officials, for their support and valuable inputs. Also, thankful to UGC for providing financial assistance during this period.

http://dx.doi.org/10.3837/tiis.2014.05.013

1. Introduction

The growth of attackers started with Denial of Service (DoS), where the attackers exploited the vulnerabilities of the protocols, which were not actually developed with security in mind. So, the attackers used these vulnerabilities to exploit the protocol in an illegitimate way, to their advantage. There are of two types of attack's namely, active and passive [1]. The concentration is on active attacks. The flooding requests from the attacker are active attacks. Flooding attacks, like the UDP and ICMP, affect the network bandwidth. The flooding attack TCP SYN flooding affects both the bandwidth and the personal computer recourses [2]. These attempts are very weak to generate, since an attacker with a raw socket can create a number of attacks by just altering the flag of the TCP packet. The masquerading IP addresses used for hiding the identity of the source address to deceive a victim destination, or the number of victim destinations, are known as IP spoofing [3]. IP spoofing is used as a tool by the attackers to create a number of spoofed attack packets with the help of a single personal computer.

Distributed Denial of Service (DDoS) remains a serious problem in the internet even today. It does not allow legitimate users to access the resources provided by the servers. There are many types of attacks, such as the SYN flood attack, ACK flood attack, IP Fragmentation, Distributed Reflected Denial of Service, Teardrop attack and Smurf attack, associated with the denial of service, which are created using TCP vulnerabilities. Attackers may target the bandwidth, storage capacities and processing ability of the network. Malicious attackers use the three way hand shake between the client and the server, and the limitations of the half-open connections of the TCP. Several methods have been proposed to mitigate such attacks [4-5].

As the attackers moved towards the Distributed Denial of Service (DDoS), they made use of a number of compromised systems to launch distributed attacks, making use of spoofing which, in turn, motivated the attackers to launch the reflection attack. The Distributed Reflection Denial of service (DrDoS) attacks are generated, using the spoofing technique, as shown in **Fig. 1**. The reflection attacks create a devastating effect, since the legitimate TCP server is utilized by the attackers to target a victim server.

The two main advantages for the attackers launching these DrDoS attacks are:
1. Anonymity
2. Amplification

Anonymity is provided to the DrDoS attacks via spoofing the actual attacker as the primary target, and the attacker posing to the primary target as being directly attacked by the victim servers.

Amplification is provided by the multiple victim servers; the attacker's initial request yield's a response that is larger than what was transmitted, thus increasing the attack bandwidth.

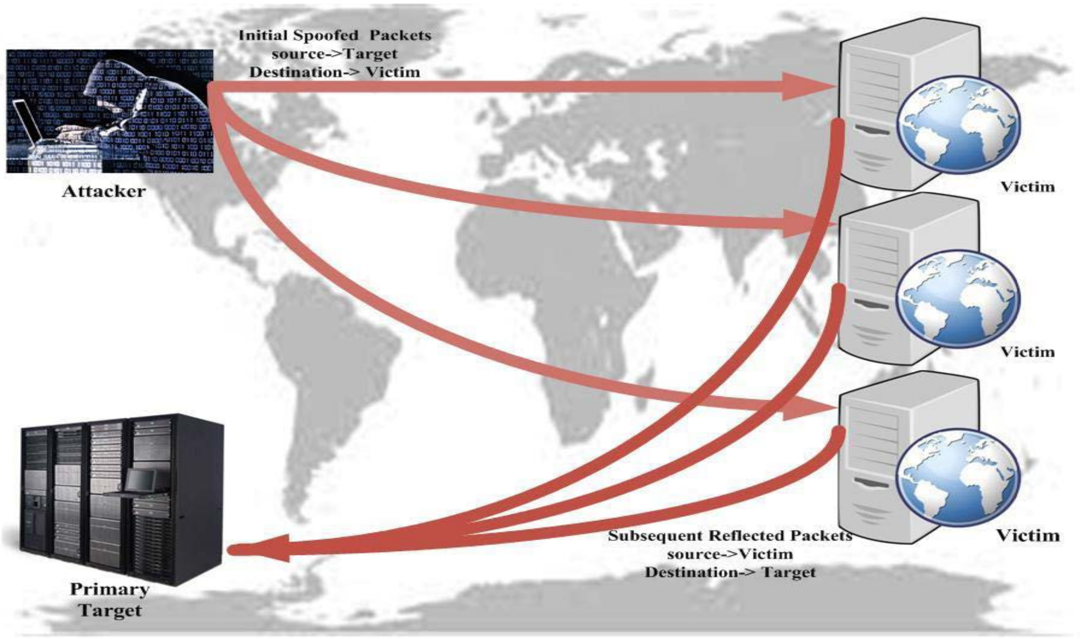

Fig. 1. DrDoS attacks scenario

The existing system finds the availability of the client. Based on the client's availability, decisions are made. But the legitimacy of the source client is only half satisfied while checking the availability of the client source. The client source by default does not ensure the security feature of non-repudiation. These reflection/amplification attacks are performed with the aid of the spoofing techniques [6]. It is very essential to counter the spoofing of the IP addresses, which forms the basis of the creation of these DrDoS attacks. In this study, the proposed system not only finds the availability of the client, but also keeps intact the non-repudiation property.

In Section 2, we review the existing work in DDoS and IP spoofing mitigation. Section 3 addresses the various scenarios in which the proposed Efficient Spoofed Flooding Defense works. We describe the result and analysis of the real time experiments carried out in the proposed ESFD based system, and the attacked system in Section 4.

2. Related Work

The DrDoS is the combination of both DDoS and IP spoofing attacks. So, the related work deals with a combination of both the defense techniques.

The solutions to problems like the DDoS and IP spoofing, detect, mitigate and filter the DDoS attacks. The DDoS & DrDoS attacks, generated using the IP spoofing technique, are detected and defended, using the techniques at various levels like the source, destination and intermediate levels [7].

The Source level prevention schemes like Ingress/Egress filters [8], D-WARD [9-10], MULTOPS [11], MANAnet's Reverse Firewall [12] were utilized. These prevention schemes, however suffer from the following issues. First, the sources of the attacks can be distributed in different domains, making it difficult for each of the sources to detect and filter the attack flows accurately. Secondly, it is difficult to differentiate between the legitimate and attack traffic near the sources, since the volume of the traffic may not be big enough, and it typically aggregates at points closer to the destinations. Finally, the motivation for the deployment of the source-based mechanisms is low, since it is unclear who (i.e., customers or service providers) would pay the expenses associated with these services. Hence, pure source-based mechanisms are not efficient and effective against DDoS flooding attacks.

The intermediate router level defense scheme, proposed by Shu Zhang and Partha Dasgupta [13] mooted a router based solution, in which the flooding is stopped at the router level itself. They made use of hardened routers, which provide digital signatures and encryption. Even though this arrangement provides more secure and private communication between the routers involved, a tremendous amount of implementation complexity is required. Establishing hardened routers increases the cost of the scheme. In addition, the last router is critical as it decrypts the initial packet; thus, a single point of failure can consequently create a less reliable information system.

Destination level defense schemes like StackPi, CAPTCHA, IP puzzle, and HOP count filtering, are implemented for the detection of attack packets; the attacks are considerably reduced using these schemes. In the StackPi scheme, the router may mistakenly drop legitimate packets, if it simply drops all the packets with an “attack path” marking [14].

The server requires a proof of work from the client, before committing its resources to the client. The challenge response methods, like IP puzzling and Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) are used to find the legitimate user [15-17]. This graphical Turing test is posed by an overlay node, to decide on whether a client is allowed to get connected to a trusted approved location, so as to access the web server using Secure Overlay Services (SOS) [18-19], and WebSOS, to protect a web service against DDoS attacks. Clients expend their resources to solve a “crypto-puzzle” and submit a proof of the solution as an embedded signal (capability) within each packet. The limitation is with regard to the asymmetric computing power for the end users. Since computational puzzles give advantage to end-users with faster CPUs, mobile devices which have less power cannot receive the services.

Hop Count Filtering by Jin, G., Wang, H., and Shin, K. G, is a victim based solution, and it is based on the fact that the number of hops between the source and the destination is indicated by the TTL field in the IP packet [20-21]. The drawback of the Hop count filtering scheme is that a network failure shows a change in the hop count, or the attacker may try to spoof its IP address from the same hop count distance, while the IP puzzle scheme suffers from the overhead of sending, receiving and matching the puzzles. The Economic Denial of Sustainability (EDOS) shield is virtual firewalls (VF) and verifier cloud nodes (V-Nodes). The virtual firewalls work as filter mechanisms based on white and black lists that hold the IP addresses of the originating nodes. And the verifier cloud nodes update the lists based on the results of the verification process. This scheme can be easily exploited by attackers, by initially sending a genuine request followed by malicious attacks [22-23]. In 2011, Kavisankar *et al.*, proposed a TCP probing method to determine the availability of the customer with less implementation complexity [24-25]. The drawback is that, it only checks for the availability of the client, but does not verify if the packets are from the original client’s IP, i.e., checking the Non-Repudiation feature.

The proposed method is provided at the destination level because of its feasibility and accuracy, as discussed in the above destination level defense scheme. The existing state-of-the-art schemes suffer from issues such as detection accuracy, implementation feasibility and computational complexity. How these demerits are overcome in the proposed scheme, called Efficient Spoofed Flooding Defense (ESFD) to avoid the DrDoS attack, will be discussed in the following section.

3. Efficient Spoofed Flooding Defense Scheme

3.1 Efficient Spoofed Flooding Defense scheme Architecture

In this paper, we propose a scheme, called Efficient Spoofed Flooding Defense (ESFD), to defend against a DrDoS attack which is generated due to a spoofed flooding attack. This new scheme involves a two level checking of the client's request. The components of two level checks, are probing and time stamping. The existing solutions to defend against TCP SYN flood and DrDoS attacks involve checking only the client's availability using probing, to confirm that the incoming packet is from a legitimate source. But, this does not ensure the security feature of non-repudiation, i.e., whether the packet was originally sent by the client or by an attacker who sent a packet by spoofing the IP address with the legitimate client's IP address. In the proposed scheme, the introduction of time stamping in the packet will take care of the security feature of non-repudiation, which is not available in the existing schemes. To defend against the DDoS, in the case of IP spoofing too, our proposed solution of two level checking, functions very efficiently. The architecture of the proposed scheme is as follows.

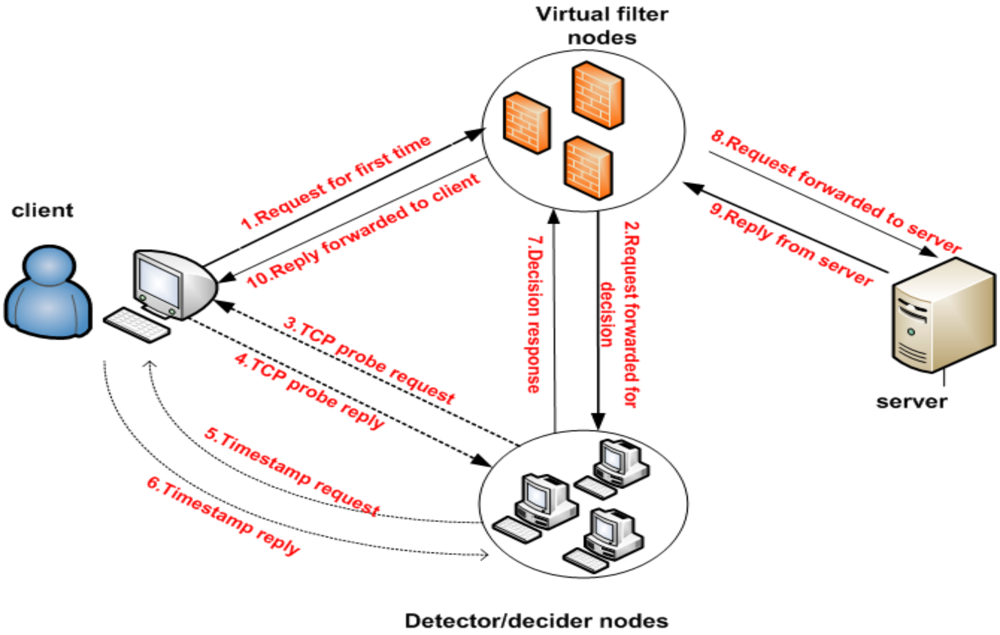

Fig. 2. Architecture of the proposed ESFD Scheme

The architecture consists of four components, which are:

1. The Client: The Client initiates the request to the server to get his services. This may be either legitimate or malicious.
2. The Server: This provides the requested service to the client, based on his request. The request must be identified as to whether it is legitimate or attack. The Server always provides the service upon receipt of the request; hence, the attack request must be avoided in the server.

3. Virtual filter: This is one of the important components of the proposed architecture.
The main function of the virtual filter is to allow only the legitimate packets and deny the attack packets. This is done based on the white listed and blacklisted IP addresses, which will be discussed later.
4. Detector/ Decider nodes: This is another important component of the proposed architecture.
The function of the detector/decider node is to detect and decide whether an IP address is to be in the white list (WL) or the blacklist (BL) or White migrated Black (WmB) list.

Assumptions:
1. Clients in the network are clock synchronized, and with no network failure.
2. The time stamp is enabled in the packets.

The architecture of our proposed scheme to protect the critical server infrastructure from the DrDoS is shown in **Fig. 2**. The architecture is designed to have both the virtual filter and the verifier nodes, which reside in a single node to reduce the network latency. The verifier node is just a piece of code, which performs the verification step on behalf of the virtual filter. So both the components are in the same personal computer. If they are in different systems, the network latency would be an overhead and processing each request will be time consuming and inefficient.

3.2 Two level checking of the proposed scheme

Probing: This is the first level of checking for client availability. TCP probing is used to check the client's availability. If the client is available, then the second level of checking is initiated.

Time stamping: This is the second level of checking for non-repudiation. In this stage, the detector/decider node queries the client to send the timestamp of the last sent packet to the destination server.

If the timestamp replied by the client and the timestamp of the packet received by the detector/decider node are the same, then the client is legitimate, and the IP is placed in the white list; else it will be placed in the black list.

This is the general working principle of the proposed architecture. How this principle is incorporated for various attack scenarios, will be discussed in the following sections.

3.3 Description of attack scenarios

Four scenarios are taken to determine the correctness. In the first scenario, the legitimate user tries to access the server for the first time, and makes an entry in the white list of the virtual filter. In the second scenario, the attacker who acts like a client and tries to access the server for the first time, sends an illegitimate request and makes an entry in the black list. In the third scenario, the attacker spoofs the source IP address of a legitimate client when the client is not connected to the network. In the fourth scenario, the legitimate client whose IP is spoofed, is currently connected to the network, and turns out to be a DrDoS attack on the client. In the third and fourth scenarios, entry is either in the white or black list, based on the decision taken by the decider node. Now we shall discuss the above mentioned scenarios, one by one.

3.3.1 Scenario 1- Legitimate client's request for the first time

When the client sends a legitimate request for the first time, the following sequence of actions takes place.

The client's first request is sent to the virtual filter. Since the request is a first time one, the IP address of the client will not be available in the known white and black IP address lists of the virtual filter. Hence, the virtual filter verifies the client by forwarding the request to the detector/decider node. The detector/decider node does two levels of checking. The first level of checking includes TCP probing, which is achieved by introducing a challenge number in the probe request, sent by the detector/decider node. If the client's acknowledgement contains the same challenge number, the availability of the client is ensured. But the client may not be legitimate also. Hence, the second level of checking is initiated. In the second level check for non-repudiation, the time stamp is introduced at the request. If the same time stamp is acknowledged correctly, non-repudiation is ensured. As the attacker is not in the exact client location, the time stamp acknowledged must be different. Hence, the attacker and the legitimate client will be easily identified. This ensures both availability and non-repudiation, which are not considered in any other existing work.

When the client is found to be legitimate, the detector/decider node informs the virtual filter to make an entry in the white list of the IP addresses. Hence, the client's request is forwarded to the server. The server's response is again forwarded back to the client. This trusted communication between the client and the server will be continued. This scenario is illustrated in **Fig. 2**.

3.3.2 Scenario 2- Attacker's illegitimate request for the first time

The attacker, who acts like a client, now sends a request to the server. Like the previous scenario1, the request is first handled by the virtual filter. The virtual filter finds that the IP address is not in the known list of the IP addresses. Hence, it forwards the request to the detector/decider node. The detector/decider node now deploys the two level checks. In the considered scenario, an illegitimate request is sent by the attacker. Hence, the two level checks fail, and an entry is made by the virtual filter in the black list of the IP addresses. So, further requests from this IP address are denied by the virtual filter.

3.3.3 Scenario 3- Generation of spoofed flooding attack using offline client's IP address

In this scenario, the client's IP address is legitimate, and it has an entry in the white listed IP addresses in a virtual filter's list. Now, the attacker sniffs until the client is disconnected from the network, and initiates the spoofed flooding attack. Since the IP address is in the white list, according to the existing scheme, the request will be forwarded to the server. This existing scheme uses a huge amount of the server's resources and bandwidth, which in turn, causes denial of service to legitimate requests.

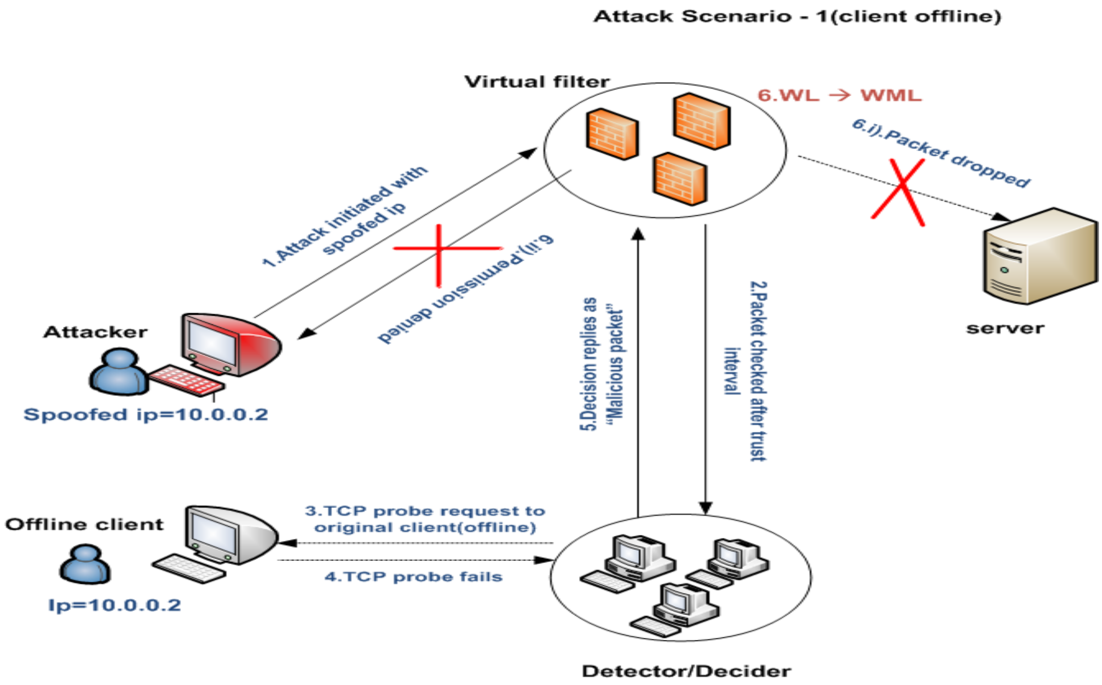
Attack Scenario - 1(client offline)
Virtual filter
6.WL → WML
6.i).Packet dropped
1.Attack initiated with spoofed ip
6.ii).Permission denied
Attacker
Spoofed ip=10.0.0.2
5.Decision replies as "Malicious packet"
2.Packet checked after trust interval
server
Offline client
Ip=10.0.0.2
3.TCP probe request to original client(offline)
4.TCP probe fails
Detector/Decider

Fig. 3. Generation of spoofed flooding attack using offline client's IP address

To resolve this issue, according to our proposed solution, even if the IP address is in the white list we subject it to periodic checks. We term this periodic interval as the trust interval.For example, if the trust interval is 10 requested packets, after 10 requests, the two-level check is initiated once again. So, even if the attacker tries to spoof using a white listed IP address, the attack will be detected only at the end of the trust interval. This is at a negligible level.

In the case of the existing solutions, the attack is detected or undetected only after a certain threshold of the resources of the victim is exhausted. In this scenario, the legitimate user is currently not a part of the network. Hence, after the end of the trust interval, we employ the client availability check using TCP probing. Since the client is not part of the network, the TCP probe fails and the IP address is migrated to the WmB list, and the packets from this IP are denied until the end of the trust interval. The malicious packets of the attacker are dropped now. Meanwhile, if the client gets connected to the network to send the legitimate packets, its packets are denied for a maximum period of the trust interval, and then are allowed. But in the existing literatures, the legitimate clients are also denied the service. This is illustrated in the **Fig. 3**.

3.3.4 Scenario 4- Generating DrDoS attack using online client's IP address

In this scenario of **Fig. 4**, the client's IP address is legitimate, and it has an entry in the white listed IP addresses in the virtual filter's list. Now, the attacker initiates the spoofed flooding attack of the client connected to the network. Since the IP address is in the white list, according to the existing solutions, the request will be forwarded to the server. This uses the server's resources and bandwidth causing denial of service to the legitimate requests.

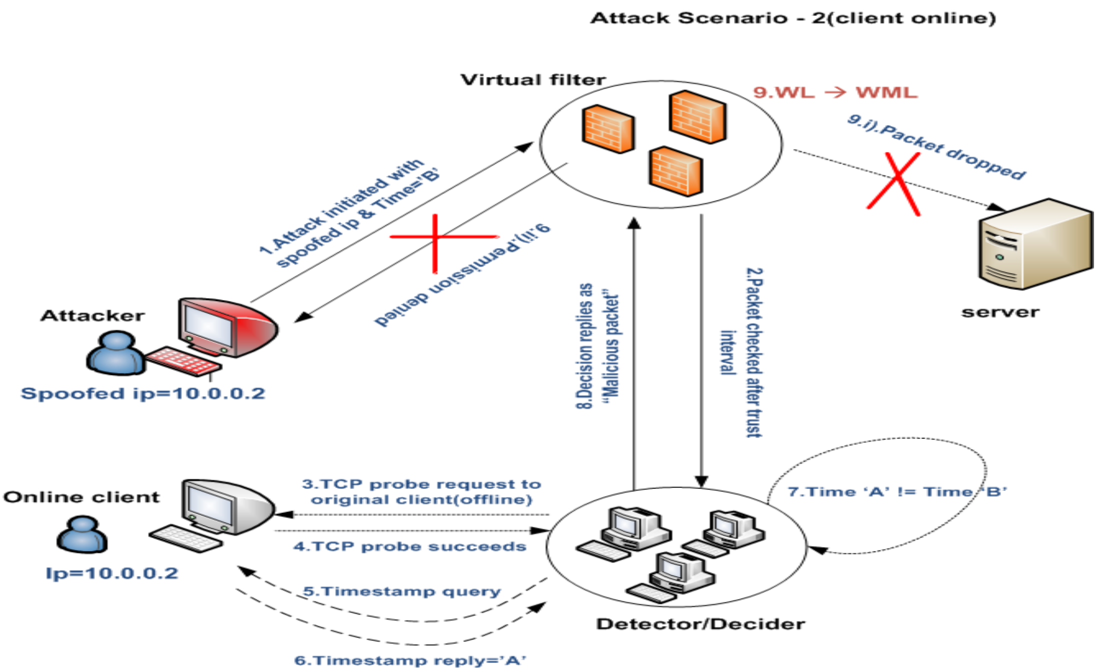

Fig. 4. Using online client's IP address to generate DrDoS attack

Consider the same trust interval as 10; i.e., the IP addresses in the white list are checked for every 10 packets. So, when the attacker tries to spoof its IP address as that of the legitimate client, the DDOS attack is detected by the end of the trust interval. The client's availability is checked using the TCP probe. Even if this is successful, there is no guarantee that the IP address is legitimate.

So, a second level of checking is performed. The client is queried for the timestamp of the last packet sent to the destination server. Let the timestamp be A. Let the timestamp in the received packet with the same IP address be B. If A=B, then the packet sent is legitimate; else, the packet is sent by a malicious attacker. The attack is detected, and the IP address is migrated from the white list to the WmB list.

In this case, we divide the clients into high and low priority clients, based on the probability of the attacks from their network. The known clients are called the high priority clients, and the unknown clients fall under the low priority category. Higher priority packets are allowed, without the two level checks, when they are in the White List. If the two level checks fail for the low priority network, packets are dropped, and the IP addresses in these packets are added to the Black List. The packets in the high priority network are periodically checked; if they do not satisfy any one of the checks, they are added to the WMBL; else, to the BL. A threshold of 10 is given as a privilege to the high priority network, because of the lesser probability of an attack. But in the case of low priority, the threshold is 5 packets, which is less than the high priority, because of the higher probability of attack. The threshold is fixed based on the number of experiments conducted.

In this section, different scenarios were discussed, with respect to the proposed ESFD. The following section explains the experimental setup and real time formation of the DrDoS attack. The performance of the proposed ESFD scheme is compared with that of the attacked system, and the results are discussed.

4. Experimental Results and Analysis

The previous section used the two level checks, to ensure the critical server's ability to withstand the DrDoS attack caused by the spoofed flooding packets. In this section, the performance of the proposed system is evaluated experimentally. The critical server's performance of the proposed system is compared with the attacked system's performance. The experiment is conducted using the experimental setup consisting of Eight Institutes/Universities (sites) situated in different geographical locations, working collaboratively on a Smart and Secure Environment (SSE) project, as shown in **Fig. 5**. The Institute (Site 2) (Anna University) is one among the sites connected through the 2 Mbps Multi Protocol Label Switching (MPLS) Virtual Private Network (VPN) cloud. Each site maintains a web, mail, DNS, and proxy servers. The proposed system has been conceived, designed and deployed at Site 2 to monitor the real time traffic and to filter the malicious traffic. DrDoS attacks are performed by the attackers using Spoofed flooding packets from the other collaborating sites of the SSE project. The primary memory used in this experimental setup is the 3 GB DDR.

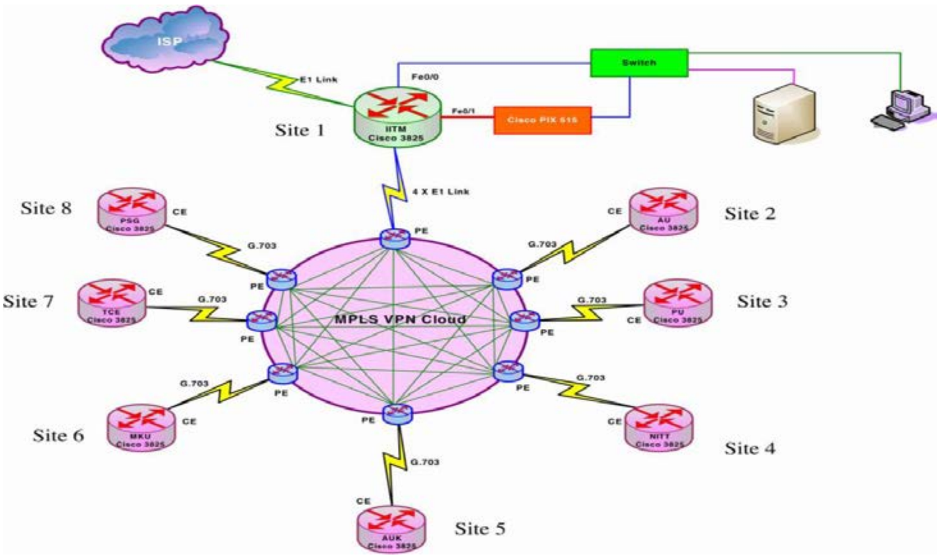

Fig. 5. SSE experimental environment for the defense of DrDoS using the Proposed ESFD Scheme during the generation of DrDoS attacks

The attacks are performed from sites 5&6 (i.e. Alagappa University (ALU), and Madurai Kamaraj University (MLU)) respectively. They try to generate a DrDoS attack by flooding the spoofed packets with the IP address of site 2, Anna University (AU), which happens to be the primary target of the DrDoS attackers in this instance. Sites 1, 3, 4, 7&8 (i.e. Indian Institute of Technology Madras (IITM), Pondicherry University (PU), National Institute of Technology Tiruchirappalli (NITT), Thiagarajar College of Engineering (TCE), and PSG College of Technology (PSG)) are the victim servers. These victim sites reflect their responses in a collaborated manner. These collaborated flooded packets flood the primary target (Site 2-Anna University (AU)). The measure of CPU, memory, network bandwidth utilization and the number of attacks and legitimate packets reaching, are logged. The virtual filters are implemented on the victim server. The information of the attack traffic is logged and analyzed, to measure the impact of the Spoofed DDoS attacks.

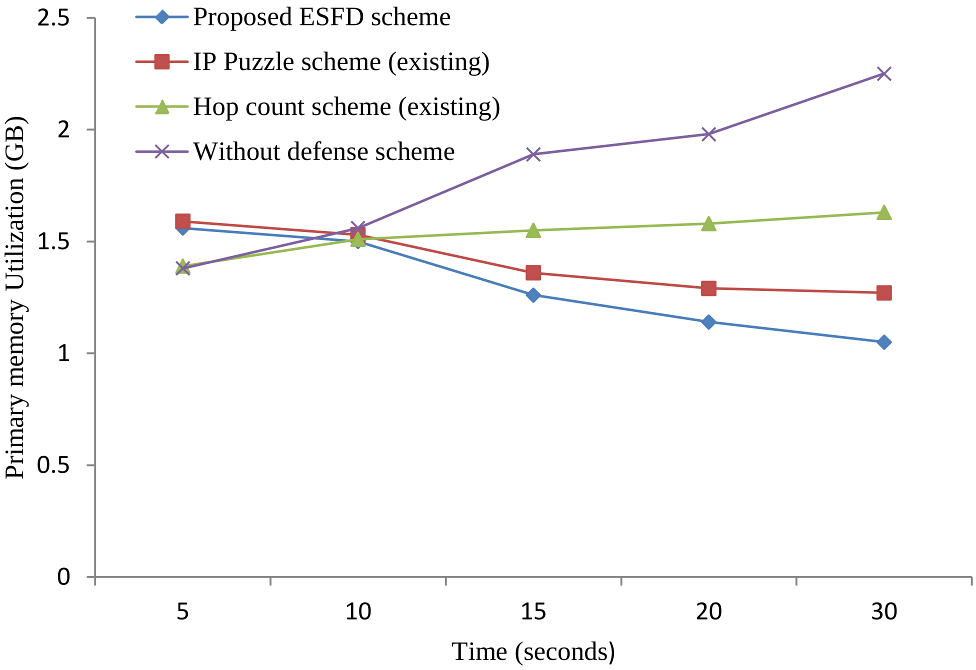

Fig. 6. Comparison of the primary memory utilization of the primary target

Optimizing the primary memory usage with the two level check scheme: **Fig. 6** shows the graph of the time (in seconds) and the primary memory usage (in GB), and compares the proposed ESFD scheme with the various existing ones, viz, IP Puzzle, Hop count, and without defense schemes, for the attacked system. Generally, in an attacked system, the number of packets flooded by the attackers causes more memory usage in the system. The performance of the system will be greatly impeded if more memory is used. In general, the increase in time will increase the number of attacked packets, which in turn, increases the memory usage.

Initially, in the time span of 5 seconds, the proposed scheme consumes 52% (i.e. 1.56 GB) of the total memory. But an attacked system consumes only 46% (i.e. 1.38 GB). This drop in percentage is initially due to the computational running overhead of the proposed scheme. In the same time period, the existing IP puzzle scheme consumes 53% (i.e. 1.59 GB) due to the computational overhead caused by the generation of IP puzzles, and the existing Hop count scheme consumes only 47% (i.e. 1.39 GB) due to simply monitoring the TTL value present in the packet. These results show that the proposed ESFD scheme initially performs 1% better than the existing IP puzzle scheme, and 5% poorer than the existing Hop count scheme, in utilizing the primary memory. This drop in the primary memory performance in the proposed ESFD scheme is only temporary. When the time increases, automatically the performance of the proposed ESFD scheme increases, and it shows better performance over the other existing schemes, as it uses the two level check scheme to detect and defend against the attacked packets.

In the time span of 10 seconds, the attacker's system consumes 1.56 GB of the total memory, and the proposed ESFD consumes only 1.5 GB memory. This shows a 2% improvement over the attacker system. In the time span of 15 seconds, the proposed ESFD scheme shows 4%, 10% and 21% improvement in memory usage over the existing IP puzzle, Hop count, and without defense schemes, respectively. Similarly, the proposed ESFD scheme shows an improvement in memory usage by 5%, 15% and 28% over the existing IP puzzle, Hop count, and without defense schemes respectively, in the time span of 20 seconds. These experimental results show that the proposed system initially consumes more memory as it uses the two level checks, which does not happen in the attacked system. The above graph plotted, using real time experimental data, very clearly illustrates the improvement in memory usage. During the 30 seconds of the time period, the proposed ESFD scheme's implemented system consumes only 35% of the total memory, but the attacked system consumes 75% of the memory resources. In the same time period, the existing IP puzzle and Hop count schemes consume 42% and 54% of the memory resources. This shows that the proposed scheme greatly improves the memory performance by up to 40%, leaving 65% of the resources free in the memory over the attacked system, and 7% and 19% over the existing IP puzzle and Hop count schemes respectively.

Due to the computational overhead caused by the generation and verification of IP puzzles in the existing IP puzzle scheme, and failure in mitigating the attacked packets when increasing the time in the existing Hop count scheme, it uses more primary memory compared to the proposed ESFD scheme. As the proposed scheme works on the two level check methods viz, TCP probing on the request packets, and the timestamp on the packets, it considerably reduces the flooding attack packets, which in turn, decreases the memory usage. Our objective is to improve the performance of the system by reducing the memory usage. This is achieved by using the proposed ESFD scheme over the various other existing schemes for defense. The experimental results also prove this betterment, which is illustrated through the graph plotted using these results.

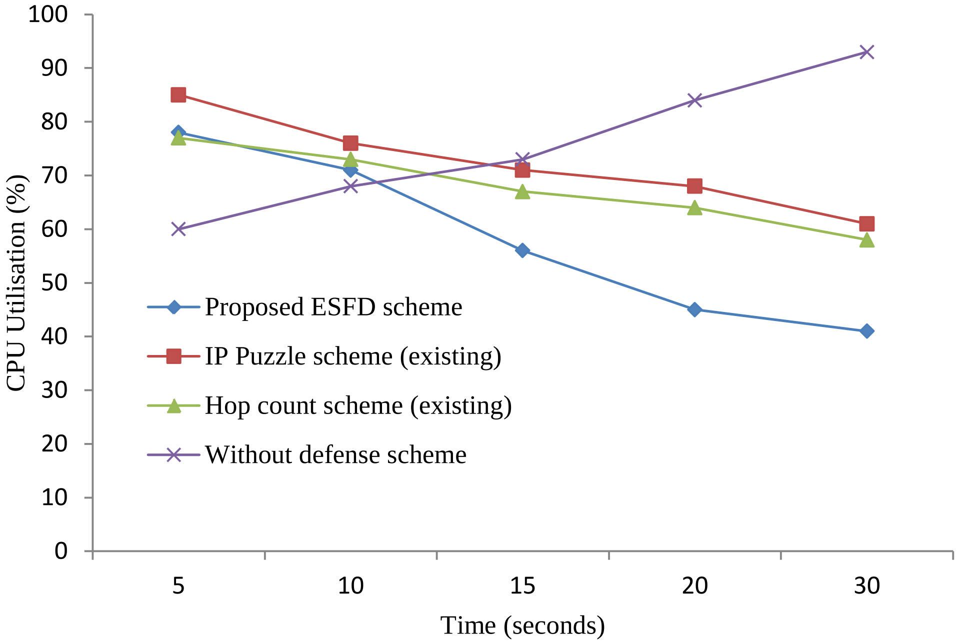

Fig. 7. Comparison of CPU utilization of the primary target

Optimizing the CPU utilization, using the two level checks scheme: **Fig. 7** shows the graph of the time (in seconds) and the CPU utilization (in GHZ) in the proposed ESFD based system, and compares the proposed ESFD scheme with the various existing IP puzzle, and Hop count schemes, and the attacked system. Generally, in the attacked system, more packets are flooded by the attackers, which cause more CPU usage in the system. Similar to the increased memory utilization, more CPU utilization of the system will also greatly impede the performance of the system. When the time increases, the number of attacked packets also increases; this in turn, increases the CPU usage.

Initially at 5 seconds, the proposed scheme based system utilizes 78% of the total CPU resource of 2.2 GHz, while the attacked system consumes only 60%. Due to the overhead caused by the two level check procedure of the proposed ESFD scheme, the CPU utilization is increased considerably, when compared to the attacked system. For the same 5 second time period, the existing IP puzzle scheme utilizes 85% of the CPU, which is 7% more than the proposed scheme, while the Hop count scheme utilizes only 77% of the total CPU which is 1% less compared to the proposed scheme. When time increases, the performance of the proposed system increases over the attacker's system. In 10 seconds, the attacked system utilizes 68% of the CPU resource, which is 3 % less than that of the proposed system. When time increases, the proposed system's performance is drastically improved over the attacker's system, and other existing systems for defense. The improvement in CPU utilization of the proposed system is 15%, 10% and 17 % over the existing IP puzzle, Hop count and without defense schemes, with respect to the time periods of 15s, 20s and 30s. Similarly, in 30s time, the proposed ESFD based system utilizes only 41% resource of the CPU, but the attacked system uses 93 % of the CPU resource, which is 52 % higher than the proposed system. Similarly, when compared to the IP puzzle and Hop count schemes, the proposed ESFD scheme shows 20% and 17% improvement in CPU utilization.

These experimental results show that the proposed system initially consumes more CPU resources as it uses the two level checks, but this does not happen in the attacked system, which does not use the two level checks. When the time increases, the proposed scheme based system performs better when compared to the attacker system and the other existing schemes for defense. The graph plotted using real time experimental data, very clearly illustrated the improvement in CPU usage. Our objective is to improve the performance of the system by reducing the CPU usage. As the proposed scheme works on the two-way check methods, like timestamp on the packets and the TCP probing the request packets, it considerably reduces the flooding attack packets, which in turn, decrease the CPU usage. Though the other existing schemes for defense like IP puzzle and Hop count schemes reduce the CPU utilization, this reduction is comparatively lesser than that in the proposed ESFD scheme.

Initially in 5 seconds, the proposed scheme and all other existing schemes consume almost 57% of the total network resources, i.e., 57 Mbps bandwidth. In spite of the presence of the computational overhead, the proposed ESFD based system performs as well as the attacked system, where there is no two level checks procedure. When the time increases, the proposed system improves the bandwidth utilization. This is clearly proved though the experimental results plotted in the graph. When the time increases, the proposed scheme increases the bandwidth over the existing IP puzzle, Hop count, and without defense schemes by 8%, 4% and 55% respectively, in the 20 seconds time period. Similarly, in 30 seconds, the proposed system utilizes 29% of the total bandwidth; but for the same set up, the attacked system utilizes 96%, which is much higher than that of the proposed ESFD based system. In the same time period, the existing IP puzzle, and Hop count utilize 38% and 35% of the total bandwidth, which is 9% and 6% more than that of the proposed ESFD scheme.

This experimental result shows that the proposed system initially consumes more bandwidth as it uses the two level checks; when the time increases the proposed scheme based system performs well, when compared to the attacker and other existing schemes. The graph plotted using the real time experimental data, very clearly illustrated the improvement in the bandwidth usage.

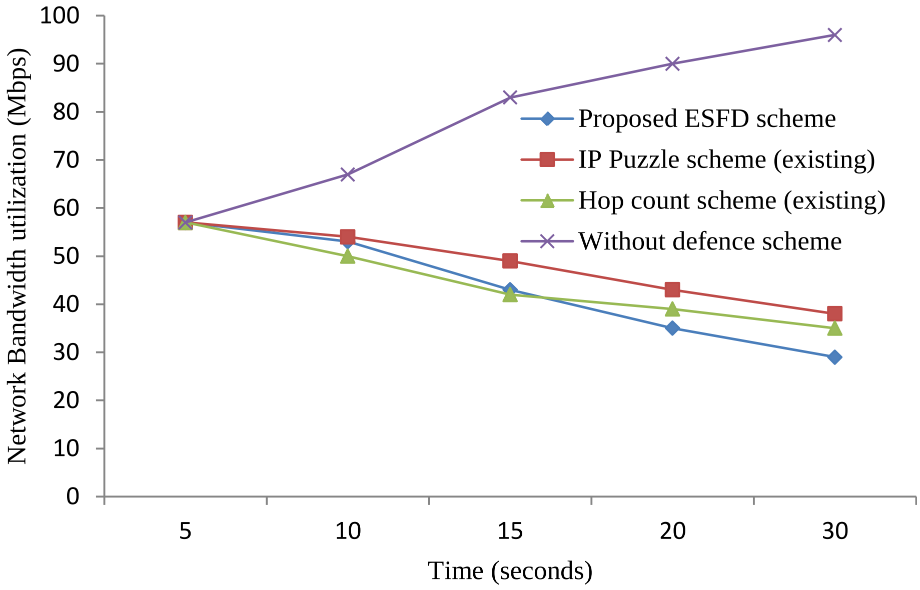

Fig. 8. Comparison of network bandwidth utilization of the primary target

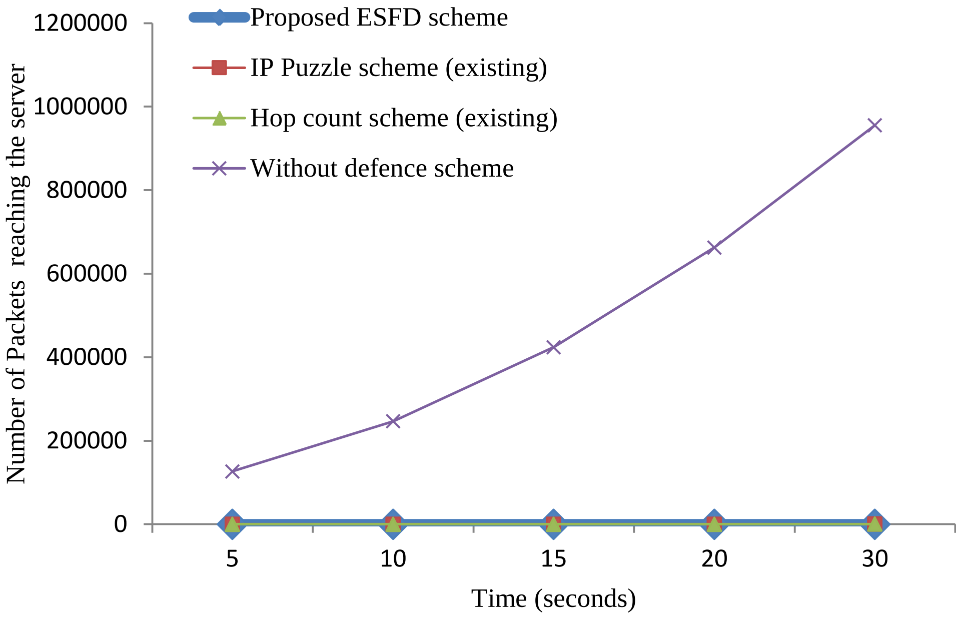

Fig. 9. Comparison of the spoofed flooded packets reaching the primary target

Increasing the accuracy of the legitimate packets reaching the server using two level check schemes: **Fig. 9** shows the graph of the time (in seconds) and the number of packets reaching the server (in numbers) in the proposed ESFD scheme based system, and the other existing schemes. The number of flooded packets reaching the victim system and the spoofed packets to the primary target increase the use of resources like memory, CPU and network bandwidth. The additional usage of the resources will greatly impede the performance of the victim and the primary target. As the proposed ESFD scheme works on the two level check methods, like the timestamp on the packets and the TCP probing the request packets, it considerably reduces the flooded attack packets, which in turn, decreases the network bandwidth usage. The graph plotted using the real time experimental data very clearly illustrated the improvement in the drop of the attack packets in the proposed ESFD scheme implemented system, and the existing IP puzzle and Hop count scheme implemented system along with the attacker's system. From the graph shown in **Fig. 9**, the attacked system shows very poor performance.

Though the attacked packets try to be clubbed with the legitimate packets, the proposed ESFD based system rejects almost all the attacked packets over the attacked system, which does not use the ESFD scheme. From the real time experimental results, we can conclude that the ESFD scheme based system initially detects 55 % of the legitimate packets, and as the time increases its detection rate also increases up to 99.2%. But the attacked system never rejects the attacked packets, which consume more memory usage, CPU utilization speed and bandwidth. This is shown in the above figure. As the proposed system rejects almost all the attacked packets, and tries to send only the legitimate packets, the line in the graph falls between 29 and 250 (which is much smaller than that of the attacked packets), and as the attacker system does not reject the attacked packets, the line falls between 126602 and 955741. To differentiate the proposed ESFD scheme from the other existing IP puzzle and Hop count schemes, a separate graph is drawn in **Fig. 10**.

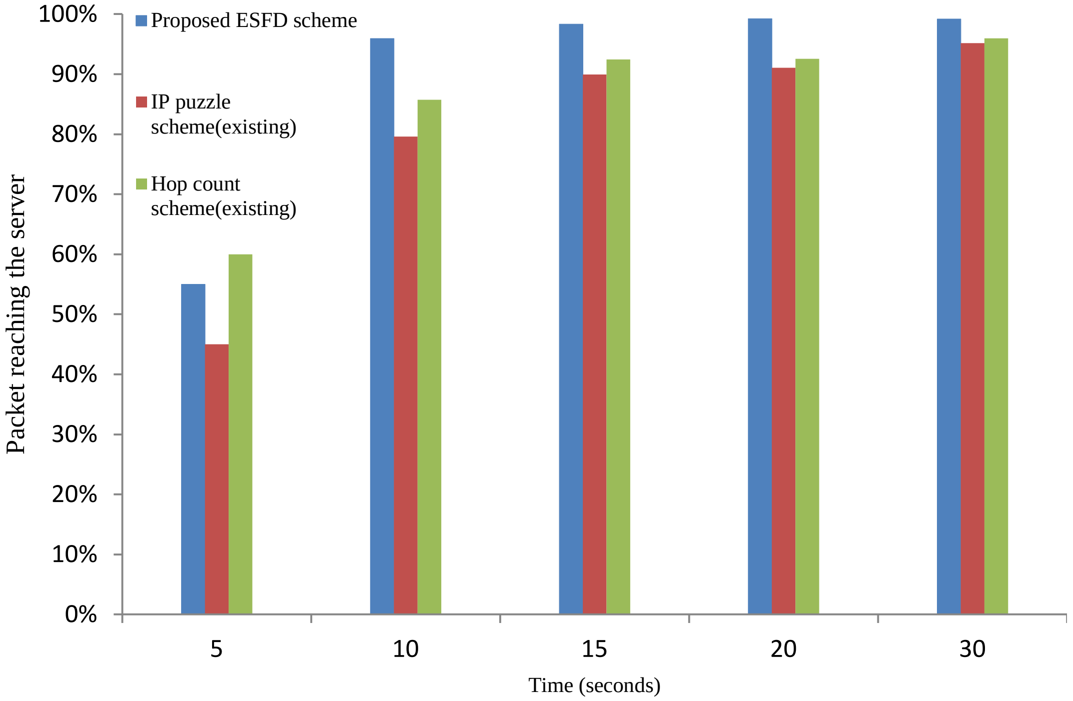

Fig. 10. Comparison of the spoofed flooded packets reaching the primary target

From **fig. 10**, in the 10 second's time span, the proposed ESFD scheme shows 16% and 10% improvement in sending legitimate packets to the victim server, over the existing IP puzzle and Hop count schemes.

5. Conclusion

In this paper, we have proposed a new scheme, named Efficient Spoofed Flooding Defense, to mitigate the DrDoS attack. This scheme for the spoofed flooding attack detection was done in two parts. The probing is done to check the availability of the client. The timestamp enhances the client with the non-repudiation feature. It is assumed that there is clock synchronization and no network failure. On that point, four scenarios are discussed in this work, which show how the proposed method provides significant good performance, especially in the scenario where the attacker spoofs the source IP address of a legitimate client, when the client is not connected to the network. Another most important scenario addressed in this paper is that, when a legitimate client whose IP is spoofed, is currently connected to the network, and turns out to be a DrDoS attack on the client.

As the proposed ESFD scheme works on the two level check methods, like the timestamp on the packets and the TCP probing the request packets, it considerably reduces the flooding attack packets, which in turn, decreases the memory usage, CPU utilization and network bandwidth usage. The real time experimental results showed the efficiency of our proposed ESFD scheme, by increasing the performance of CPU up to 40%, the memory up to 52% and the network bandwidth up to 67%. Also, the ESFD scheme based system increases the detection and defence rate up to 99.2%. Thus, the proposed ESFD scheme performed well in various scenarios to improve the memory usage, CPU utilization and network bandwidth, by mitigating the DrDoS attack.

References

[1] J. Mirkovic and P. Reiher, "A taxonomy of DDoS attack and DDoS defense mechanisms," *ACM SIGCOMM Computer Communications Review*, vol. 34, no. 2, pp. 39-53, April 2004. Article (CrossRef Link)

[2] R. K. C. Chang, "Defending against flooding-based distributed denial of service attacks: A tutorial," *Computer J. IEEE Commun. Magazine*, vol. 40, no. 10, pp. 42-51, 2002. Article (CrossRef Link)

[3] Toby Ehrenkranz, Jun Li, "On the state of IP spoofing defense," *ACM Transactions on Internet Technology (TOIT)*, vol. 9, no. 2, pp.6:1-6:29, 2009. Article (CrossRef Link)

[4] C. Douligeris and A. Mitrokotsa, "DDoS attacks and defense mechanisms: classification and state-of-the-art," *Computer Networks*, vol. 44, no. 5, pp. 643-666, April 2004. Article (CrossRef Link)

[5] Saman Taghavi Zargar, James Joshi and David Tipper, "A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, 2013. Article (CrossRef Link)

[6] T. Peng, C. Leckie and K. Ramamohanarao, "Survey of network-based defense mechanisms countering the DoS and DDoS problems," *ACM Computer Survey*, vol. 39, no. 1, April 2007. Article (CrossRef Link)

[7] Noureldien A. Noureldien and 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," *International Journal of Networks and Communications*, vol. 2, no. 3, pp. 33-37, 2012. Article (CrossRefLink)

[8] P. Ferguson and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which

employ IP Source Address Spoofing," *RFC 2267*, 2000. Article (CrossRef Link)

[9] J. Mirkovic, G. Prier, and P. Reiher, "Attacking DDoS at the Source," in *Proc. of 10th IEEE International Conference on Network Protocols (ICNP'02)*, Washington DC, USA, 2002. Article (CrossRef Link)

[10] J. Mirkovic, G. Prier and P. Reihe, "Source-End DDoS Defense," in *Proc. of 2nd IEEE International Symposium on Network Computing and Applications*, April 2003. Article (CrossRef Link)

[11] T. M. Gil and M. Poleto, "MULTOPS: a data-structure for bandwidth attack detection," in *Proc. of 10th Usenix Security Symposium*, Washington, DC, pp. 2338, August 13–17, 2001. Article (CrossRef Link)

[12] MANAnet, "The Reverse Firewall: Defeating DDOS Attacks Emanating from a Local Area Network," *DDoS White Papers Cs3, Inc*, pp.1-5, 2014. Article (CrossRef Link)

[13] Shu Zhang and Partha Dasgupta, "Hardened networks: incremental upgrading of the Internet for attack resilience," in *Proc. of The 12th International Conference on Computer Communications and Networks*, pp.595-598, Oct 2003. Article (CrossRef Link)

[14] Yaar, Abraham, Adrian Perrig and 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. Article (CrossRef Link)

[15] Feng, Wu-Chang and Ed Kaiser. "System and methods of determining computational puzzle difficulty for challenge-response authentication," *U.S. Patent Application 13/050,123*, filed March 17, 2011. Article (CrossRef Link)

[16] L. V. Ahn, M. Blum, N. J. Hopper and J. Langford, "CAPTCHA: using hard AI problems for security," in *Proc. of 22nd international conference on Theory and applications of cryptographic techniques (EUROCRYPT'03)*, Eli Biham (Ed.). Springer-Verlag, Berlin, Heidelberg, pp.294-311. 2003. Article (CrossRef Link)

[17] Ma, M., "Mitigating denial of service attacks with password puzzles," *Information Technology: Coding and Computing, ITCC*, 2005. Article (CrossRef Link)

[18] A. Stavrou, D. L. Cook, W. G. Morein, A. D. Keromytis, V. Misra and D. Rubenstein, "Websos: An overlay-based system for protecting web servers from denial of service attacks," *the International Journal of Computer and Telecommunications Networking*, vol. 48, no.5, pp.781–807, August 2005. Article (CrossRef Link)

[19] A. D. Keromytis, V. Misra and D. Rubenstein, "SOS: Secure Overlay Services," in *Proc. of Conference on Applications, technologies, architectures, and protocols for computer communications SIGCOMM'02*, vol.32, no.4, pp.61-72, October2002. Article (CrossRef Link)

[20] Nisha H Bhandari., "DDoS Attack Prevention In Cloud Computing Using Hop Count Based Packet Monitoring Approach," *International Journal of Advanced and Innovative Research (IJAIR)*, vol. 2, no.4, 2013, pp.954-956. Article (CrossRef Link)

[21] H. Wang, C. Jin and K. G. Shin, "Defense Against Spoofed IP Traffic Using Hop-Count Filtering," *IEEE/ACM Transactions on Networking*, vol. 15, no. 1, pp.40-53, February 2007. Article (CrossRef Link)

[22] M. H. Sqalli, F. Al-Haidari and K. Salah, "EDoS-Shield - A Two-Steps Mitigation Technique against EDoS Attacks in Cloud Computing," *Fourth IEEE International Conference on Utility and Cloud Computing*, pp.49-56, 2011. Article (CrossRef Link)

[23] S. Ranjan, R. Swaminathan, M. Uysal and A. Nucci, and E. Knightly, "DDoS-Shield: DDoS-Resilient Scheduling to Counter Application Layer attacks," *IEEE/ACM Transactions on, Networking*, vol. 17, no. 1, pp. 2639, February 2009. Article (CrossRef Link)

[24] L. Kavisankar and C. Chellappan, "CNoA: Challenging Number Approach for uncovering TCP SYN flooding using SYN spoofing attack," *International Journal of Network Security & Its Applications (IJNSA)*, vol. 3, no. 5, pp.191-202, 2011. Article (CrossRef Link)

[25] L. Kavisankar and C. Chellappan, "T-RAP: (TCP Reply Acknowledgement Packet) a Resilient Filtering Model for DDoS Attack with Spoofed IP Address," *Communications in Computer and Information Science*, vol. 197, pp.138-148, 2011. Article (CrossRef Link)

L. Kavisankar is a Ph.D student in the Department of Computer Science and Engineering at Anna University,Chennai, India. He did his B.E Computer Science and Engineering from Easwari Engineering College affiliated to the Anna University in 2007 and M.E Computer Science and Engineering from SSN College of Engineering affiliated to the Anna University in 2009, Chennai, India. He has published 6 papers in reputed International Journals and Conferences. His current research is on mobile IPv6 and network security.

Dr. C. Chellappan is professor in the Department of Computer Science and Engineering at Anna University, Chennai, India. He received his B.Sc. degree in Applied Sciences and M.Sc in Applied Science–Applied Mathematics from PSG College of Technology, Coimbatore, affiliated to the University of Madras, in 1972 and 1977. He received his M.E and Ph.D degrees in Computer Science and Engineering from Anna University in 1982 and 1987. He is currently the Dean of the College of Engineering, Guindy, Anna University, Chennai. He was the Director of Ramanujan Computing Centre (RCC) for 3 years at Anna University (2002–2005). He has published more than 70 papers in reputed International Journals and Conferences. His research areas are computer networks, Distributed /mobile computing and soft computing, software agent, object oriented design and network security.

Dr. P. Sivasankar is Assistant Professor in the Department of Electrical & Electronics and communication Engineering, National Institute of Technical Teachers Training and Research, Government of India, Ministry of Human Resource Development, Chennai, India. He received his B.E degree in the Electronics and Communication Engineering from the University of Madras in 2001. He did his M.E in Applied Electronics from the Anna University in 2005. He has received Ph.D degree in Computer Science from Anna University in 2013. He has published more than 20 papers in reputed International Journals and Conferences.

His research areas are Mobile Adhoc Networks And Wireless Sensor Networks.

Srinivas Avireddy is a Software Engineer working for PayPal, EbayInc.,Chennai, India. He received his B.Tech in Information Technology from the Madras Institute of Technology, Anna University in 2013. He has published 4 papers in reputed International Journals and Conferences. His current research is in the areas of Big Data Analytics, Data Mining and Machine Learning.

N. Ashwin Karthi is a student in the Department of Computer Science and Engineering at College of Engneering, Guindy, Anna University, pursuing his final year in B.E Computer Science and Engineering. His current research is in the areas of Machine Learning, Natural Language Processing and Distributed Systems.