Advanced Random Time Queue Blocking with Traffic Prediction for Defense of Low-rate DoS Attacks against Application Servers

R.Kavitha[1], Dr.G.Padmavathi[2]

[1]Department of Computer Science, Avinashilingam Institute of Home Science and Higher Education for Women
[1]Department of Computer Science, Sri Krishna Arts and Science College, Coimbatore, Tamilnadu, India.
[2]Department of Computer Science, Avinashilingam Institute of Home Science and Higher Education for Women, Coimbatore, Tamilnadu, India.

Abstract: Among many strategies of Denial of Services, low-rate traffic denial-of-service (DoS) attacks are more significant. This strategy denies the services of a network by detection of the vulnerabilities in performance of the application. In this research, an efficient defense methodology is developed against low-rate DoS attack in the application servers. Though, the Improved Random Time Queue Blocking (IRTQB) technique can eliminate the vulnerabilities in the network and also avoiding the attacker from capturing all the server queue positions by defining a spatial similarity metric (SSM). However, the differentiation of the attack requests from the legitimate users' is not always efficient since only the source IP addresses and the record timestamp are considered in the SSM. It was improved by using Advanced Random Time Queue Blocking (ARTQB) scheme that employed Bandwidth utilization of attacker in IRTQB to detect the DoS attack that normally consumes a huge number of resources of the server. However, this method becomes ineffective when the attack consumes more network traffic. In this paper, an efficient detection technique called Advanced Random Time Queue Blocking with Traffic Prediction (ARTQB-TP) is proposed for defining SSM which contains, Source IP, timestamp, Bandwidth between two requests and the difference between the attack traffic and legitimate traffic. The ARTQB-TP technique is utilized to reduce the attack efficiency in 18 different server configurations which are more vulnerable to the DoS attacks and where the attacks may also have a chance to improve its effectiveness. Experimental results show that the proposed system performs better protection of application servers against the LRDoS attacks by solving its impacts on any kind of server architectures and reduced the attack efficiencies of all the types of attack strategies.

Keywords: Low-rate denial-of-service, Spatial Similarity Metric, IRTQB, Traffic prediction, Attack efficiency, Application server.

1. Introduction

An application server is referred as software structure which offers both conveniences for producing web applications and a server environment for functioning them. This software structure includes the comprehensive service layer model. This performs as a group of elements available to the software developer via an application programming interface which is described by the platform itself. In web applications, the elements are presented in similar running environment as web servers whereas in Java applications, the server acts as a comprehensive virtual machine. It is mostly used for providing middleware services for security and maintenance including with data access and perseverance.

Denial of service (DoS) is a web based attack aiming to make critical resource unavailable to legitimate users [1]. The DoS attack is an event to prevent the legitimate users from accessing the network services. That is to make a machine or network resource unavailable to its legitimate users, by temporarily or indefinitely interrupting the services of a host connected to the Internet. The attacker sends excessive messages asking the network or server to authenticate requests that have invalid return addresses. As the server will not be able to find the return address of the attacker it waits before a moment before closing the connection. Once the server closes the connection, the attacker sends more authentication messages and process of authentication and server wait will begin again, keeping the network or server busy that will cause effects on the legitimate users server need.

There are different ways that a DoS attacks can be implemented. Flooding the network to reduce the legitimate network traffic, disrupting the connections between the user and the server, blocking certain range of users and disrupting the state of the information of the users are the basic ways that are implemented. DoS generates high inconsistent traffic rate and thus detection became easier, but then low rate attacks [2] came into existence in which the DoS is achieved in low traffic scenarios.

Low-rate DoS attack (LRDoS) are new and significant type of DoS attacks where the attacker sends a sudden cloud of well-timed packets and increments the retransmission timeout for only certain TCP flows as they may be lost at intervals. As these traffic disturbances occur during the expiration times, it reduces the efficiency in detection of this attack. Many techniques have been presented in the recent past for the detection of low rate DoS but most of the techniques performed with low efficiency. Shrew and reduction of quality (RoQ) attacks which is a type of DoS increases the detection complexity in the system.

Defense techniques [3] for low rate DoS attacks were developed against application servers. Among these techniques Improved Random Time Queue Blocking (IRTQB) was achieved less attacker efficiency than other techniques to protect LRDoS attacks against application servers. Hence, we explore traffic prediction including with IRTQB to prevent LRDoS attacks against application servers.

2. Related Works

Erwin Adi,et al [4] demonstrated the DoS attack in http servers. The paper showed how the attacks can be launched and be detected. The attacker's victim memory resources gradually deplete at a certain rate by the low rate DoS attack in the server. It was also possible to launch stealthy DoS attacks by sending packets with rules. The paper showed that a single attacker can successfully attack 12 servers. The DoS attacks could not be made stealthier by adding time delays between the attack packets.

Wu Zhi-juna et al [5] proposed an approach for the detection of LRDoS attacks based on Duffing oscillator in chaos systems. It does it by adopting the technology of digital signal processing (DSP) that considers LRDoS attack as a small signal and TCP traffic as background noise. The Duffing oscillator is used to detect LRDoS attacks in normal TCP traffic. LRDoS attacks can be detected through diagram of the chaotic state, period, and pulse width of LRDoS attacks can be estimated. However, the computational complexity of this system is high.

Yonghong Chen et al [6] initially pre-processed the network traffic using the simple linear AR model, and then it generated the prediction of network traffic. It assumed that the prediction error behaves chaotically and used Chaos theory to analyze it. This work proposed a novel network anomaly detection algorithm (NADA) to detect the abnormal traffic in the network. A neural network is trained to detect DDoS attacks that accurately and effectively detect the DDoS attacks.

Xinlei Ma et al [7] proposed anomaly detection method based on the Tsallis Entropy and Lyapunov exponent. The entropy between Source IPs and Destination IPs were compared by analysis of the rate of exponent separation. A variation of Lyapunov exponent was proposed based on entropy. The proposed approach outperformed all the techniques and the proposed Exponent Separation Detection Algorithm can detect the DDoS attack efficiently. The paper combined the effect of source IPs and destination IPs in network traffic that helped for efficient detection of the attacks. However it lacks well structuredDoS datasets.

Xianliang Jiang et al [8] proposed a method with flows trust values for the protection against DoS attacks. The proposed scheme employed the flow trust to safeguard legitimate flows. Here a router monitors the network flows and calculates flows trust values for the relevant queue management. The attacker flows would be of lower trust values while legitimate flows would be higher values. Proposed scheme improved the throughput and delay in DoS attacking scenarios which was practical and effective to secure networks. However, the scheme involves computational burden to the router in the network.

Alan Saied et al [9] proposed a trained Artificial Neural Network algorithm to detect DDoS attacks based on characteristic patterns of genuine traffic from DDoS attacks. Different DDoS attacks were then exposed while normal traffic was through the network. Then the datasets were collected, pre-processed and then prepared to train the algorithm. Proposed detection mechanism was then integrated with Snort-AI which was tested against known and unknown DDoS attacks. The proposed approach produced higher detection accuracy and used in various stages of attacks.

Rishie Sharma et al [10] investigated the popular Apache HTTP Server software and identified the weakness of the server. The details about the persistent connection feature handled by the server were discussed. Then an attack simulator which changed the weakness of the attack has been proposed and developed for effective detection of the attack in the server. The attack was then studied with spectral analysis for examining the effectiveness in the detection of the attack in the server.

EsraaAlomari et al [11] investigated the Botnet-based DDoS attacks and detected on the application layer. The details about Botnet based DDoS attacks on web server were studied. The occurrences and problems of Botnet based DDoS attacks were discussed. The financial issues through these attacks in popular industries and government websites were illustrated. This study was further carried out for computing an optimal solution for these attack issues.

Maryam Tanha et al [12] developed discrete event simulator for analyzing DoS attacks. The attributes and mechanisms for this simulator were developed and the security in specific DoS attack was analyzed. In-depth analysis was also accomplished through this discrete event simulator against HTTP low rate DoS attacks. This discrete event simulator based analysis was then included for investigating other security problems.

Mohit Sharma et al [13] presented data mining technique to detect low rate DoS attack. The software based technique is developed for detecting LDoS attacks. This software was implemented on the server for constantly monitoring the data flow in the network. The pre-processing methods such as naïve bayes and apriori were described and generation of data was explained. This method was also accountable for attack traffic.

In this paper, the DoS detection scheme named Improved Random Time Queue Blocking (IRTQB) and Advanced Random Time Queue Blocking is analyzed to determine the detection efficiency. The analysis results show that the IRTQB performs better than existing methods however the IRTQB also suffers from limitations. Particularly the differentiation between the attack requests from the legitimate users' requests is not satisfactory. Hence bandwidth utilization is included in IRTQB to develop Advanced Random Time Queue Blocking (ARTQB) scheme for effective defense of the application servers. This method also proves to be poor in detection in case of loss of traffic to legitimate users. Thus the difference between the legitimate user and the attacker is added as additional parameter to the Spatial Similarity metric to detect the attack efficiently. This paper also works to prove the proposed detection mechanism is able to defend against the DoS attack in 18 different architectures of the server that are more vulnerable and the attacks try to improve in their architecture.

3. Proposed Mechanisms

3.1 Application Server Model

Application servers are heavily vulnerable to LoRDAS attacks shown in Figure 1. There are situations are required for an application to be at risk of this kind of attacks, and several distinctive techniques might be followed by means of the attacker to deny the service of the server. The application server model considered in the LoRDAS assault is composed of the following elements (i) a service queue wherein incoming requests are placed upon their arrival at the server and (ii) one or numerous carrier modules which can be in charge of processing the requests.

The low-rate DoS assaults within the application servers relies upon on two major components of server conduct which includes the presence of deterministic styles and allowing instants concurrence with the answer instants. The defense strategies are developed based totally at the method used for lowering the performance of the attack within the servers and with none negative effect on the normal performance of the servers.

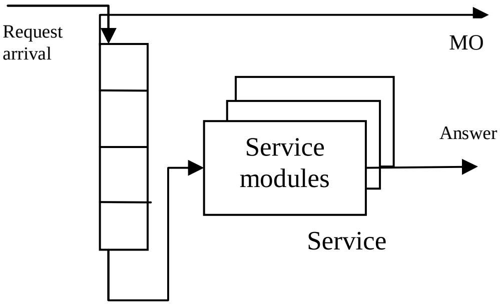

Figure 1. Application Server Model in LoRDAS Attack

3.2 Drawbacks of Existing Methodologies

Several techniques were introduced in order to detect the LRDoS attacks in the application servers, among them there were RST, RAI, RTQB, IRTQB [9] and ARTQB. Each technique has its own advantages and disadvantages. The RST detects the LRDoS attack by shifting the answer time to a place that was not always managed through the attacker and also with the help of a source of variability that was included within the server behavior. However, there is an extra delay added to the original service time, making the attacker additionally perceive an increase in service time which could be solved by adjustment of the attack parameters to synchronize the attack bursts. RAI technique performs better than RST.

However in RAI the trade off among the discount of the effect and increase of variability in the server affects the normal behavior of the server, hence to avoid this problem, RTQB is presented. RTQB reduces the impact on server by reducing the assault efficiency. It makes attackers to brief bursts of visitors that arrive around the solution instants and blocks all of the incoming requests in a time interval. This turns into a major drawback in RTQB that results in a stepped forward technique referred to as IRTQB. IRTQB employs SSM and selectively blocks the requests in order that the queue has as a minimum one loose space and the attack efficiency is decreased. The SSM has accessibility on the time stamp and the source IP of the requests. The attack efficiency further increased in ARTQB by adding one more parameter to the SSM called as the bandwidth between the two requests in the server.

However the ARTQB may become less effective if the traffic is not under control and the LRDoS attack consumes more traffic. So the SSM metric should also include the difference in the attack traffic and the legitimate traffic.

3.3 Advanced Random Time Queue Blocking with Traffic Prediction (ARTQB-TP)

When RTQB is considered, the main drawback is that it does not choose requests between the blocking intervals, but it discards all of them which may contain the requests from the legitimate user. So IRTQB [9] was presented which would discard the requests from the attacker, by analyzing the requests using spatial similarity metric called SSM that records the time stamp and the source IP from where the requests arrive. The attacker's requests will have a temporal similarity of the arrival time in a time interval around the answer instants, and also a spatial similarity, between the attack requests as they are forced to follow certain communication rules given by transport and network layer protocols.

The advanced RTQB involves the same similarity metric with a parameter included in the SSM as bandwidth between the requests.

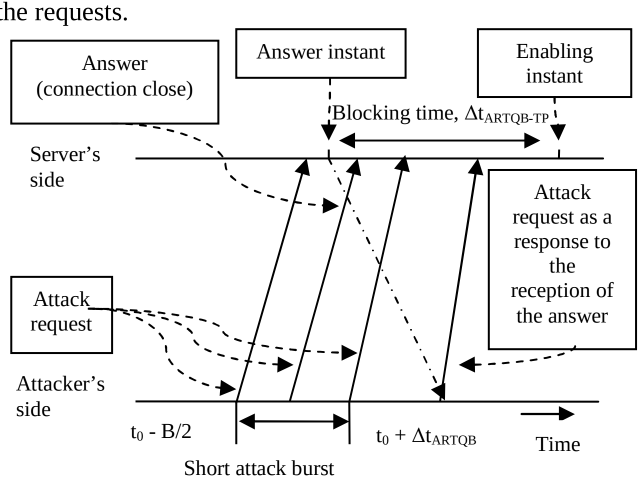

Figure 2. LoRDAS Attack when ARTQB-TP is Active

Hence a highly efficient version of RTQB with traffic prediction is proposed which is called as Advanced Random Time Queue Blocking with Traffic prediction (ARTQB -TP) as shown in Figure 2. Similar to ARTQB, a spatial similarity metric is computed considering the record timestamps, source IP addresses and the bandwidth utilization between two requests and the difference between the attack traffic and the legitimate traffic. SSM for two generic requests is computed similar to ARTQB but additionally the traffic prediction is considered.

$$SSM(A_i, A_j) = \neq_{Consecutive_{bits1}} (A_i XNORA_j) + BW(A_i, A_j) + T \quad (1)$$

Where $consecutive_{bits_1}(A_i XNORA_j)$ are number of consecutive bits set to 1 in the Bit wise XNOR of the two addresses A_i and A_j, and BW is the bandwidth between the two requests and Tis the classification threshold that defines whether the traffic is attack traffic or the legitimate traffic.

3.3.1 Algorithm for ARTQB-TP

1. Extract request from service queue
2. Processing request at service time T_s
3. Insert answer instant in a list L
4. Compute attack interval T_I
5. Send answer to users
6. Extract new request (enabling instant)
7. For every incoming request R_a
8. Record timestamp, source IP, bandwidth utilization
9. Determine T_I for R_a
10. For all requests R_b in T_I
11. SSM metric calculation by Collecting the Network Traffic packets and flow information in the network.

12. Calculate the entropies of the source IPs and the destination IPs in every time units by Tsallis entropy defined as

$$H_q = \frac{1 - \sum_{i=1}^{N} p_i^q}{q-1} \qquad (2)$$

13. The maximum entropy is defined as

$$H_q^{max} = \frac{1 - N^{1-q}}{q-1} \qquad (3)$$

14. The entropy of the observed traffic is normalized with respect to maximum entropy, H_{norm} $(0 \leq H_{norm} \leq 1)$ is defined by

$$H_{norm} = \frac{H_q}{H_q^{max}} \qquad (4)$$

15. The entropy sequence of source is H_s and that of destination isH_d, AR model is used to generate the dynamical equations to pre process the sequences.
16. $\hat{H}_s(i+1) = F[\hat{H}_s(i)]$ And $\widehat{H}_d(i+1) = F[\hat{H}_d(i)]$ are the dynamical equations.
17. Analyze the rate of exponent separation between the source IP and the destination IP by,

$$T_k = \frac{1}{t_k} \ln \frac{\hat{H}_s(k)}{\hat{H}_d(k)} \qquad (5)$$

18. Where $T_k$is used to represent the exponent separation.
19. When $T_k > 0$, it defines whether the traffic is normal or anomaly. The reason of anomaly in this case could be traffic burst or a DDoS attack.
20. When$T_k > 0$, it defines whether the traffic is normal or anomaly. The reason of anomaly in this case could be network scanning.
21. So a classification threshold value is set based on the observation of the network traffic defined by T.
22. SSM metric is derived and threshold for SSM SSM_{Th} is determined by
23. If (SSM (R_a, R_b)>SSM_{Th}) then
24. Discard R_a and R_b
25. Else
26. Insert Request in Queue
27. End if
28. End for

3.3.2 Description

ARTQB - TP extracts the requests from the service queue and processes them at the service time T_s. After completing the processing the answers are send to the legitimate users who requested them while on the other side called enabling instant, the new requests are started to process. These requests arriving at attack interval T_I are selectively chosen and those from the attackers are discarded. When the attack interval expires, new requests are accepted again and the processing begins. ARTQB maintains a list of answer instants from the beginning of the server operation.

SSM metric is calculated based on the Source IP, Bandwidth between the requests and the Traffic prediction. The Traffic prediction is calculated based on the variation of Lyapunov exponent that is used to detect anomalies in network traffic, based on entropy. A set of probabilities of each packet feature are defined for N number of packets and the Tsallis entropy is defined with q as the entropic parameter. The value of the entropy ranges from 0 to maximum value. Lyapunov exponent characterizes the rate of exponent separation between the two trajectories. Exponent separation detection algorithm is presented by taking advantage on the Lyapunov exponent and Tsallis entropy, where Tk is used to represent the rate of exponent separation which detects the DoS attack traffic and a classification threshold T is set to separate an anomalous traffic pattern from a non anomalous one.

Thus now the ARTQB-TP maintains a list containing the timestamps, source IP addresses, bandwidth utilization and the traffic prediction whether it is attack or legitimate for all incoming requests. Using the record list, for all the incoming requests the SSM is computed between the incoming request R_a and every request R_b arriving in the attack interval. If the SSM is higher than the defined threshold SSM_{Th}, both the requests R_a and R_b are discarded while in other case the incoming request is accepted. It is also noted that there is no impact in the server behaviour when the ARTQB-TP scheme is active in the server and it also reduces the attack efficiency of the LRDoS attack.

3.3.3 Reducing the LRDoS Attack Efficiency in Different Server Configurations

The proposed ARTQB-TP technique is utilized in 18 different server configurations which were highly vulnerable to this low rate DoS attack. The performance of the attack is tested by its efficiency parameters such as Availability of Service (A), percentage of available time (Tav) and traffic rate overhead (O) [10]. The settings of the attack parameters were selected as t_{on}time∈ [0.2 s, 0.7 s], Δ∈ [0.15 s, 0.35 s], Δ_r∈[1 s, 5 s] and Na equal to the number of service queue positions in the server. In all the scenarios, the attack was tested and the least value obtained for availability of service was 11.1% which is very worst such that for every 10 requests sent by the users, only one is served. The overhead value of the traffic was more than 230%. This represents a very high level of efficiency of the LRDoS attack in these configurations.

These kind of different attack configurations were tested using the proposed ARTQB-TP technique to efficiently overcome the effect of attack efficiency of LRDoS. The experimental results that are given below shows the proposed ARTQB-TP reduce the efficiency of the different attack configurations.

3.3.4 Improvements in attack strategy

The low rate DoS attack can also achieve a very high rate of efficiency. Several improvements can be adopted by the attack in order to obtain better efficiency of the attack. The improvements are described below.

(a) Attack distribution

The attack can also be executed by several distributed attack machines. Considerable number of attack machines was distributed to establish communication mechanisms between each other and also with the attacker.

(b) Spoofed Addresses

The attacker must receive responses from the server in order to gain knowledge about the time at which the outputs occur. Here spoofing is entertained if the range of spoofed addresses belongs to the same attacker machine. This machine is able to sniff the packets that are sent to the spoofed addresses. When the local area network range of addresses is large, spoofing will lead the attacker to conceal its location.

(c) Attack Request Diversification

The general mechanism for attack strategy is sending identical requests to the server. But the vulnerability consists of a timing scheme that does not depend on the nature of the request. Therefore the attacker can improve its efficiency by diversifying the attack requests to bypass possible mechanisms that detect the identical requests to the server.

(d) Maximization of the service time of the attack requests (MSTAR)

The attacker can extend the time of a connection between the server and itself by repeatedly sending requests to the server on the same connection before the time out expires. This could be done depending on the configuration of the server. This could be done by sending the requests with no limit and also for a specified number of times.

(e) Attack Strategy Optimization

The attack thread can recover a position by sending requests at a rate of $1/\Delta_r$ in case of failure to achieve seizure. The attacker optimizes the attack strategy itself to enhance the attack efficiency. The optimization technique improves the attack efficiency and ensures more control to the attacker by means of the following improvements.

(f) Inter-thread information sharing and seizure threshold strategy (ISS)

When analyzing the behavior of the attack is analyzed implemented with the basic seizures following strategy, one vital aspect becomes inevitable. Though the objective of attack thread is to capture only one position in the service queue, it may send requests to more positions during the attack. The attack in two threads occurs as seizure of one position each and when the attacker sends more than one request through first thread and it is accepted, the position captured by the second thread is lost. The first thread acquires both the positions. However initially it is found that this approach is not a disadvantage due to the fact that the main aim of the attack strategies is to maintain the captured positions. But two vital problems occur at the later stages.

The first problem is that if the attacker runs number of threads equal to the number of positions, the first thread captures more than one position while other threads enter recovery phase. At this phase, the requests are sent every Δ_r seconds and due to the recovery time, the traffic increases without any benefit to the attacker. The second major problem is that the first thread with more than one position also launches attack through one position only to provide attempted output and the other positions stay as un-attempted outputs causing loss for the attacker. These issues are resolved by introducing two new features: ITIS and ST.

(g) Inter-thread information sharing (ITIS) feature

Whenever an attack thread fails in the capture of a position, after executing a basic attack, it asks for the other attack threads for the positions that are unattended. Then it will attend it by programming a basic attack period around its predicted instant of occurrence. There are two implementation issues of ITIS. The first issue is that each attack thread will have to save the information's regarding the unattended outputs in a common position queue accessible to all attack threads. The information's stored should not be host-dependent and therefore every attack thread that seizes a position must estimate the instant of the output before inserting an attack request in the positions queue. In case of HTTP server, the attack thread should wait for the HTTP response, at t_{rec}, to calculate T_{output}. Additionally the maintenance task should be performed at the position queue and it leads to unattended positions to be eliminated.

Likewise, when an attack thread fails in one position, the already captured other positions are not included into the queue. During this situation, the thread utilizes its own output for speeding up the process. In addition to such instances, when there are no information's are in position queue, it has to wait for the recovery interval Δ_r. Thus the attack efficiency is found to be enhanced by resolving the multiple seizure problems.

The analysis of ITIS feature shows that it enhances the attack efficiency by the sharing of information between the different active attack threads.

(h) Seizures threshold (ST) feature

The problem of un-attempted outputs due to launch of attacks through only one seizure can be resolved using ST feature. ST feature allows the attacker to establish a threshold for the number of seizures to be simultaneously possessed P_0 which could be dynamically modified during execution. The number of positions captured P is continuously monitored. When $P<P_0$, then the attack behaves similar to that specified by ITIS feature. But when the $P> P_0$, the unattended output generated threads are modified to remain asleep till the condition changes.

The analysis of ST feature shows that the attack is able to reach a number of positions oscillating around the threshold which allows the attacker to generate a attack point. It is also found that above a certain threshold, the attacker cannot capture more positions.

These are the different strategies that the attacks can be improved as it increases the efficiency and also enables the attacker to have enhanced control over the execution. This kind of improvement in the attack can also be solved using the proposed ARTQB-TP method. The Results section shows the performance of the ARTQB-TP using the parameters such as Attack efficiency and Mean of Service time comparison.

4. Experimental Results

This section presents the experimental results that were performed to prove that the proposed ARTQB-TP solves the problems caused by LRDoS attack in different server configurations. The performance indicators that are considered were the Availability of Service (A), percentage of available time (T_{AV}) and traffic rate overhead (O). And then secondly the results show how it reduces the attack efficiency in different attack strategies in a server by considering Attack efficiency and Mean in System time as parameters.

4.1 Different Server Configurations

(a) Availability of Service(A)

Availability of Service A is defined as the percentage ratio between the number of user requests served by the server, and the total number of requests sent by the legitimate users. This parameter gives an idea of the service level experienced by the legitimate users. The availability of the Service in 18 different server configurations were calculated when the LRDoS attack is defined in the network. Then the same result is calculated when the proposed ARTQB-TP technique is applied on the attacked network configurations.

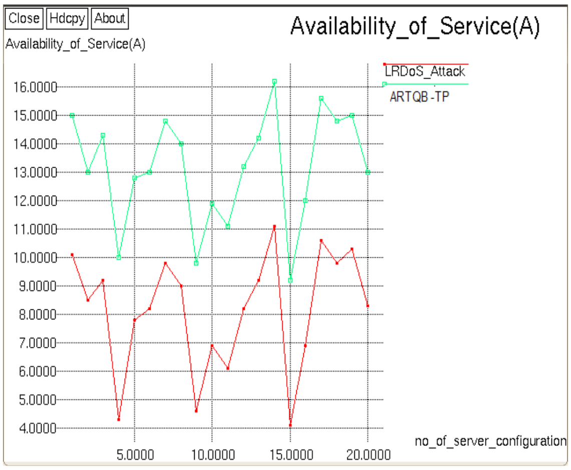

Figure 3. Comparison between Availability of Service (A) in 20 different server configurations

From the Figure 3, it is inferred that the ARTQB-TP technique improves the availability of the server to a significant level. The graph is plotted for all kinds of server configurations. For example, while considering the server configuration 10, the availability of service before was 6.9% and when ARTQB –TP is applied it has been improved to 11.9%.

(b) Traffic rate overhead (O)

The traffic rate overhead is defined as the percentage ratio between the traffic rate generated by the intruder and the maximum traffic rate accepted by the server. Figure 3 shows the comparison of the overhead value of the server.

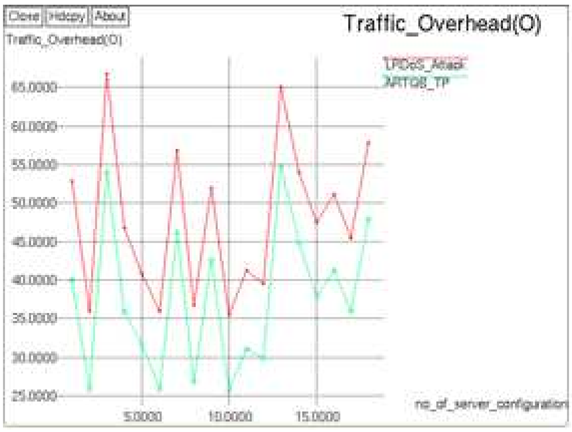

Figure 4. Comparison between Overhead (O) values in 18 different server configurations

From the Figure 4, it is inferred that the ARTQB-TP technique reduces the traffic overhead of the server to a significant level. The graph is plotted for all kinds of server configurations. For example, while considering the server configuration 12, the traffic overhead before was 39.5% and when ARTQB –TP is applied it has been reduced to 29.8%.

(c) Percentage of Available time (T_{AV})

Percentage of available time (T_{AV}) as the average time, in percentage terms, during which at least one free queue position in the server is available. The probability of a legitimate user seizing a queue position will be directly proportional to the value of T_{AV}. Figure 5 shows the comparison of the percentage of available time of the server.

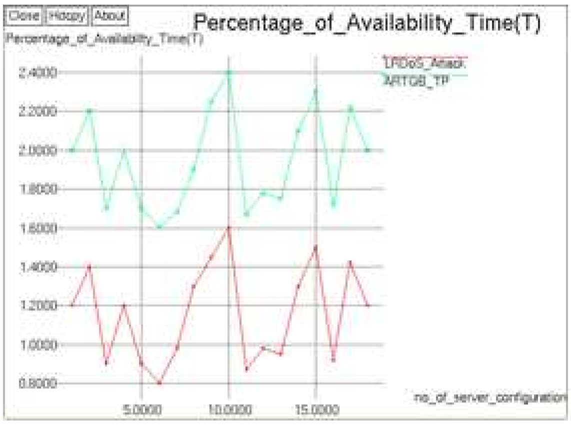

Figure 5. Comparison between percentages of availability time (T_{AV}) in 18 different server configurations

From the Figure 5, it is inferred that the ARTQB-TP technique improves the percentage of the Available time (T_{av}) of the server to a significant level. The graph is plotted for all kinds of server configurations. For example, while considering the server configuration 14, the percentage of available time before was 1.3% and when ARTQB –TP is applied it has been improved to 2.1%.

4.2 Different Attack Strategies

(a) Attack Efficiency

Attack efficiency is the percentage of service queue positions captured by the attacker over the total number of positions captured during the attack execution. The result is obtained on the basis of how attack efficiency of the different attack strategies explained before can be reduced in a server by using the proposed ARTQB - TP. The comparison is done between the IRTQB, ARTQB and the proposed ARTQB-TP techniques.

The graphs representing the attack efficiency in six different attack strategies are basic LRDoS, Attack Distribution, Spoofed address, Attack Request Diversification, Maximization of the service time of the attack requests (MSTAR) and Attack Strategy Optimization were compared for the attack efficiency.

Figure 6 shows the attack efficiency comparison of IRTQB, ARTQB and ARTQB-TP for basic LRDoS Attack that reveals that the proposed ARTQB-TP outperforms as it reduced the attack efficiency gradually. While considering the time is 8sec, the attack efficiency rate of IRTQB was 54% and ARTQB was 44.1% and when ARTQB-TP is applied it has been reduced to 32.8%.

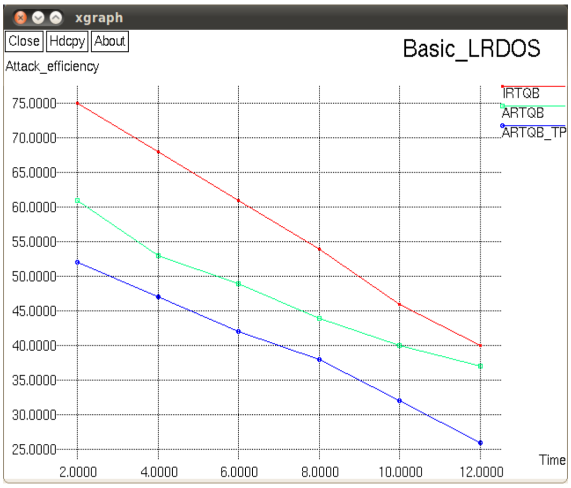

Figure 6. Comparison of Attack efficiency between IRTQB, ARTQB and ARTQB-TP techniques for basic LRDoS Attack

Figure 7 shows the attack efficiency comparison of IRTQB, ARTQB and ARTQB-TP for Attack Distribution strategy that reveals that the proposed ARTQB-TP outperforms as it reduced the attack efficiency gradually. While considering the time is 8sec, the attack efficiency rate of IRTQB was 55% and ARTQB was 48.9% and when ARTQB-TP is applied it has been reduced to 47.8%.

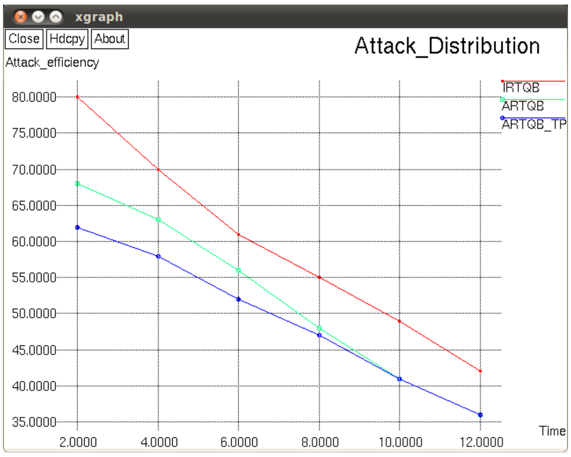

Figure 7. Comparison of Attack efficiency between IRTQB, ARTQB and ARTQB-TP techniques for Attack Distribution Strategy

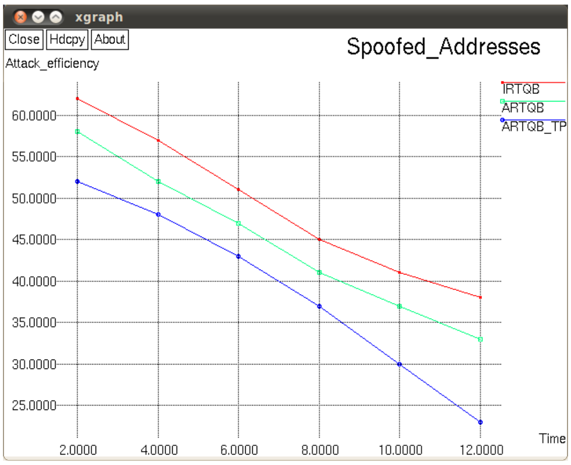

Figure 8. Comparison of Attack efficiency between IRTQB, ARTQB and ARTQB-TP techniques for Spoofed Address Strategy

Figure 8 shows the attack efficiency comparison of IRTQB, ARTQB and ARTQB-TP for Spoofed attack strategy that proves that the proposed ARTQB-TP reduced the attack efficiency gradually. While considering the time is 8sec, the attack efficiency rate of IRTQB was 45% and ARTQB was 41% and when ARTQB-TP is applied it has been reduced to 36.8%.

Figure 9 shows the attack efficiency comparison of IRTQB, ARTQB and ARTQB-TP for Attack Request Diversification strategy that proves that the proposed ARTQB-TP reduced the attack efficiency gradually. While considering the time is 8sec, the attack efficiency rate of IRTQB was 37% and ARTQB was 31% and when ARTQB-TP is applied it has been reduced to 30%.

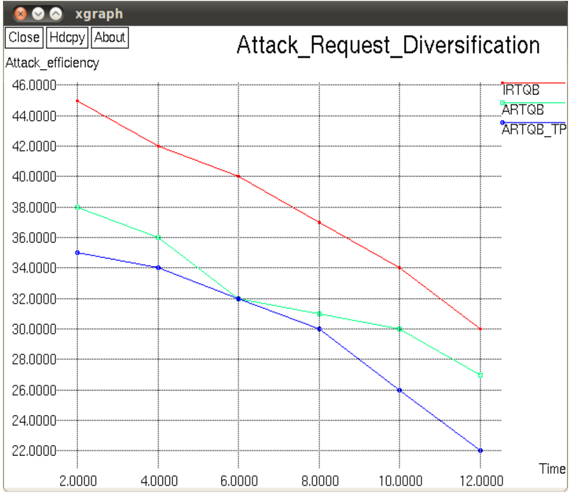

Figure 9. Comparison of Attack efficiency between IRTQB, ARTQB and ARTQB-TP techniques for Attack request Diversification Strategy

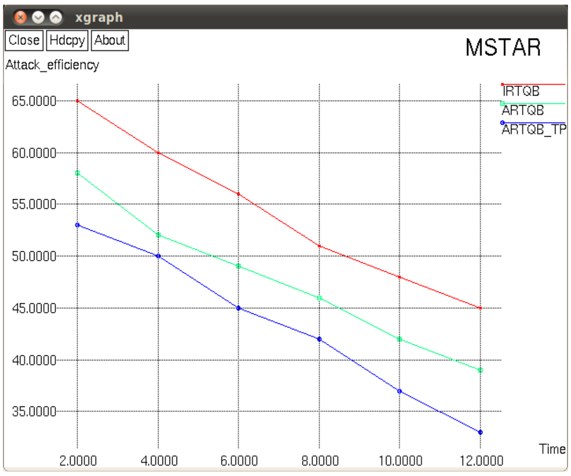

Figure 10. Comparison of Attack efficiency between IRTQB, ARTQB and ARTQB-TP techniques for MSTAR Strategy

Figure 10 shows the attack efficiency comparison of IRTQB, ARTQB and ARTQB-TP for MSTAR strategy that proves that the proposed ARTQB-TP reduced the attack efficiency gradually. While considering the time is 8sec, the attack efficiency rate of IRTQB was 51% and ARTQB was 46% and when ARTQB-TP is applied it has been reduced to 42.8%.

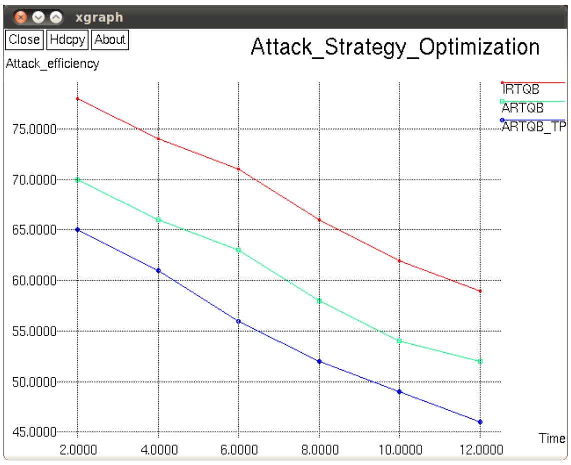

Figure 11. Comparison of Attack efficiency between IRTQB, ARTQB and ARTQB-TP techniques for Attack Strategy Optimization

Figure 11 shows the attack efficiency comparison of IRTQB, ARTQB and ARTQB-TP for Attack strategy Optimization that proves that the proposed ARTQB-TP reduced the attack efficiency gradually. While considering the time is 8sec, the attack efficiency rate of IRTQB was 36% and ARTQB was 58.5% and when ARTQB-TP is applied it has been reduced to 52.1%.

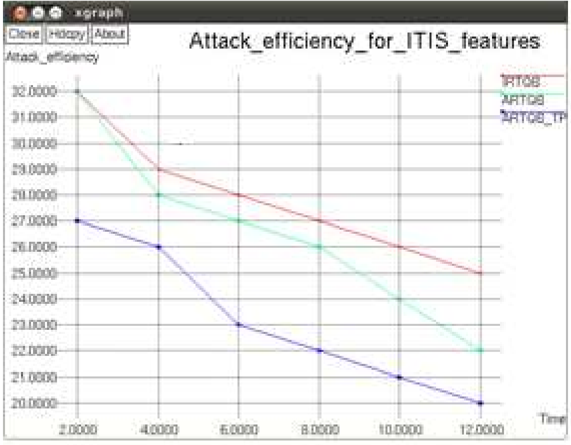

Figure 12. Comparison of Attack efficiency between IRTQB, ARTQB and ARTQB-TP techniques for ISS strategy- ITIS feature

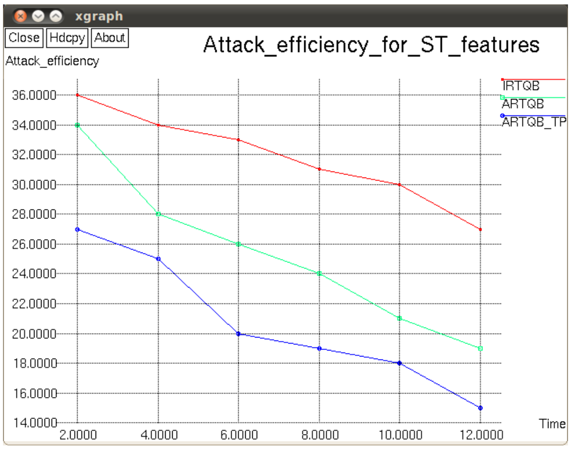

Figure 13. Comparison of Attack efficiency between IRTQB, ARTQB and ARTQB-TP techniques for ISS strategy- ST feature

Figure 12 shows the attack efficiency comparison of IRTQB, ARTQB and ARTQB-TP for ISS strategy of ITIS feature that proves that the proposed ARTQB-TP reduced the attack efficiency considerably. While considering the time is 8sec, the attack efficiency rate of IRTQB was 27% and ARTQB was 26% and when ARTQB-TP is applied it has been reduced to 22%.

Figure 13 shows the attack efficiency comparison of IRTQB, ARTQB and ARTQB-TP for ISS strategy of ST feature that proves that the proposed ARTQB-TP reduced the attack efficiency considerably. While considering the time is 8sec, the attack efficiency rate of IRTQB was 31% and ARTQB was 24% and when ARTQB-TP is applied it has been reduced to 19%.

All the results concerning the attack efficiency shows that the proposed ARTQB-TP efficiently degraded the LRDoS attack in all kinds of attack strategies along with basic strategy.

(b) Mean in System time

Mean in-system time is the time from when a request enters the server to the instant at which its corresponding answer is sent. The result is obtained on the basis of Mean in System Time of the different attack strategies explained before can be reduced in a server by using the proposed ARTQB - TP. The comparison is done between the IRTQB, ARTQB and the proposed ARTQB-TP techniques. The graphs representing the attack efficiency in six different attack strategies are basic LRDoS, Attack Distribution, Spoofed address, Attack Request Diversification, Maximization of the service time of the attack requests (MSTAR) and Attack Strategy Optimization were compared for the attack efficiency.

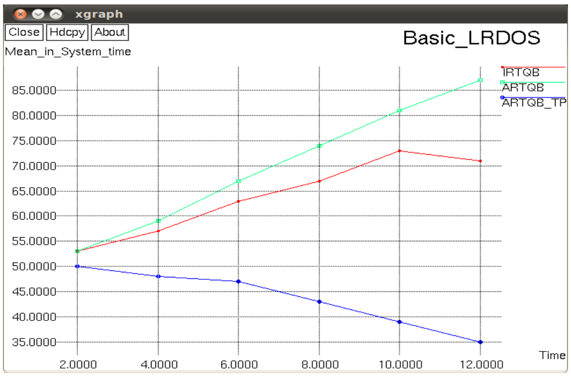

Figure 14. Comparison of Mean in System Time between IRTQB, ARTQB and ARTQB-TP techniques for basic LRDoS Attack

Figure 14 shows the Mean in System Time comparison of IRTQB, ARTQB and ARTQB-TP for basic LRDoS Attack that reveals that the proposed ARTQB-TP outperforms as it reduced the Mean in System Time efficiently. While considering the time is 8sec, the attack efficiency rate of IRTQB was 74.1% and ARTQB was 67% and when ARTQB-TP is applied it has been reduced to 43.9%.

Figure 15 shows the Mean in System Time comparison of IRTQB, ARTQB and ARTQB-TP for Attack Distribution strategy that reveals that the proposed ARTQB-TP outperforms as it reduced the Mean in System Time gradually. While considering the time is 8sec, the attack efficiency rate of IRTQB was 27% and ARTQB was 26%

and when ARTQB-TP is applied it has been reduced to 23%.

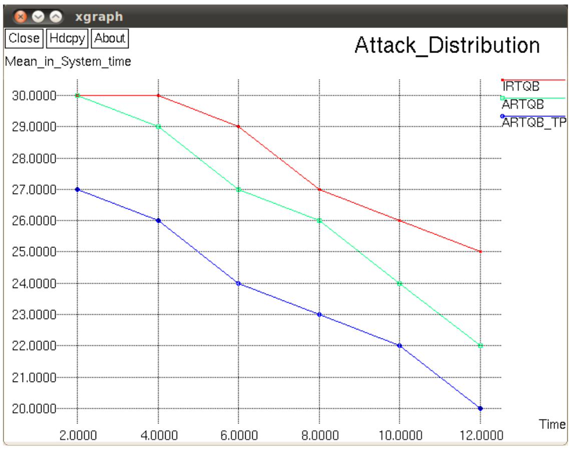

Figure 15. Comparison of Mean in System time between IRTQB, ARTQB and ARTQB-TP techniques for Attack Distribution Strategy

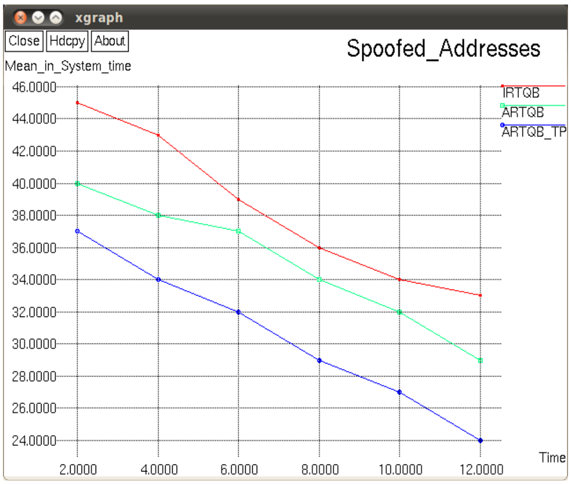

Figure 16. Comparison of Mean in System Time between IRTQB, ARTQB and ARTQB-TP techniques for Spoofed Address Strategy

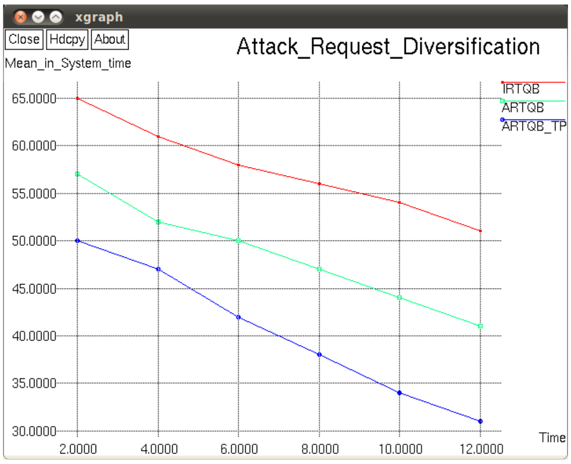

Figure 17. Comparison of Mean in System Time between IRTQB, ARTQB and ARTQB-TP techniques for Attack request Diversification Strategy

Figure 17 shows the Mean in System Time comparison of IRTQB, ARTQB and ARTQB-TP for Attack Request Diversification strategy that proves that the proposed ARTQB-TP reduced the Mean in System Time. While considering the time is 8sec, the attack efficiency rate of IRTQB was 56% and ARTQB was 47% and when ARTQB-TP is applied it has been reduced to 38.5%.

Figure 16 shows the Mean in System Time comparison of IRTQB, ARTQB and ARTQB-TP for Spoofed attack strategy that proves that the proposed ARTQB-TP reduced Mean in System Time. While considering the time is 8sec, the attack efficiency rate of IRTQB was 36% and ARTQB was 34% and when ARTQB-TP is applied it has been reduced to 29%.

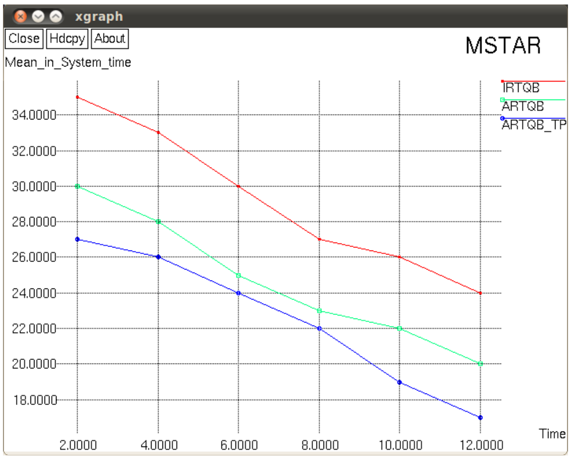

Figure 18. Comparison of Mean in System Time between IRTQB, ARTQB and ARTQB-TP techniques for MSTAR Strategy

Figure 18 shows the Mean in System Time comparison of IRTQB, ARTQB and ARTQB-TP for MSTAR strategy that proves that the proposed ARTQB-TP reduced the Mean in System time. While considering the time is 8sec, the attack efficiency rate of IRTQB was 27% and ARTQB was 23% and when ARTQB-TP is applied it has been reduced to 22%.

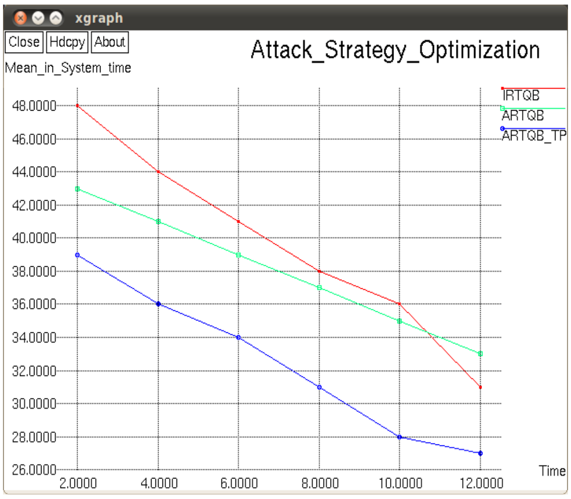

Figure 19. Comparison of Mean in System Time between IRTQB, ARTQB and ARTQB-TP techniques for Attack Strategy Optimization

Figure 19 shows the Mean in System Time comparison of IRTQB, ARTQB and ARTQB-TP for Attack strategy Optimization that proves that the proposed ARTQB-TP reduced the Mean in System Time effectively. While considering the time is 8sec, the attack efficiency rate of

IRTQB was 38% and ARTQB was 37% and when ARTQB-TP is applied it has been reduced to 31%.

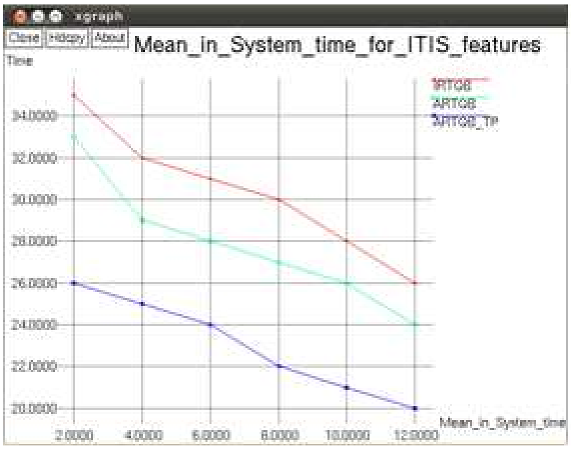

Figure 20. Comparison of Mean in System Time between IRTQB, ARTQB and ARTQB-TP techniques for ISS strategy-ITIS feature

Figure 20 shows the Mean in System Time comparison of IRTQB, ARTQB and ARTQB-TP for ISS strategy-ITIS feature that proves that the proposed ARTQB-TP reduced the Mean in System Time effectively. While considering the time is 8sec, the attack efficiency rate of IRTQB was 30% and ARTQB was 27% and when ARTQB-TP is applied it has been reduced to 22%.

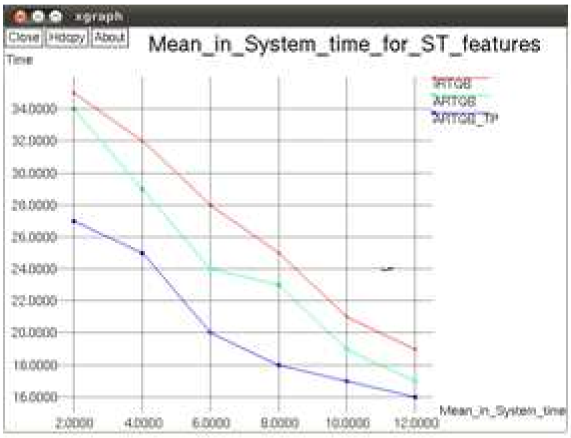

Figure 21. Comparison of Mean in System Time between IRTQB, ARTQB and ARTQB-TP techniques for ISS strategy-ST feature

Figure 21 shows the Mean in System Time comparison of IRTQB, ARTQB and ARTQB-TP for ISS strategy-ST feature that proves that the proposed ARTQB-TP reduced the Mean in System Time effectively. While considering the time is 8sec, the attack efficiency rate of IRTQB was 25% and ARTQB was 23% and when ARTQB-TP is applied it has been reduced to 18%.

All the results concerning the Mean in System Time show that the proposed ARTQB-TP efficiently degraded the LRDoS attack in all kinds of attack strategies along with the basic strategy.

5. Conclusions

Advanced Random Time Queue Blocking with Traffic Prediction (ARTQB- TP) is proposed with high intention to reduce the attack efficiency of the Low Rate Denial of Service attack (LRDoS). Detection of DoS attack is very significant in improving the server performance effectively without causing any damage to the server. Hence in this paper, maximal reduction of the attack efficiency of this attack on the server and minimizing the impact on server behaviour is assured. ARTQB - TP selectively chooses the requests during answer instants. Similarly the use of SSM with the traffic prediction along with the bandwidth utilization, source IP addresses and the record timestamp enhances the reduction of attack efficiency. Experimental results conclude that the proposed ARTQB-TP reduces the attack efficiency of the LRDoS attack as it improves the availability of the service to the legitimate user and also reduces the traffic overhead. It also battles efficiently against the improvements in that attack referred as six attack strategies along with the basic LRDoS attack by reducing the attack efficiency and the Mean in System Time.

References

[1] Iyengar, N. C. S., Banerjee, A., & Ganapathy, G, "A fuzzy logic based defense mechanism against distributed denial of service attack in cloud computing environment," International Journal of Communication Networks and Information Security (IJCNIS), pp. 233-245, 2014.

[2] Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K, "An empirical evaluation of information metrics for low-rate and high-rate DDoS attack detection," Pattern Recognition Letters, Vol. 51, pp. 1-7, 2015.

[3] Maciá-Fernández, G., Rodríguez-Gómez, R. A., & Díaz-Verdejo, J. E, "Defense techniques for low-rate DoS attacks against application servers," Computer Networks, vol. 54, No. 15, pp. 2711-2727, 2010.

[4] E. Adi, Z. Baig, C. P. Lam and P. Hingston, "Low-Rate Denial-of-Service Attacks against HTTP/2 Services," International Conference on IT Convergence and Security, pp. 1-5, 2015.

[5] Wu, Z. J., Lei, J., Yao, D., Wang, M. H., & Musa, S. M, "Chaos-based detection of LDoS attacks," Journal of Systems and Software, Vol. 86, No. 1, pp. 211-221, 2013.

[6] Chen, Y., Ma, X., & Wu, X, "DDoS detection algorithm based on preprocessing network traffic predicted method and chaos theory," Communications Letters, IEEE, Vol. 17, No. 5, pp. 1052-1054, 2013.

[7] Ma, X., & Chen, Y, "DDoS detection method based on chaos analysis of network traffic entropy," IEEE Communications Letters, Vol. 18, No. 1, pp. 114-117, 2014.

[8] Jiang, X., Yang, J., Jin, G., & Wei, W, "RED-FT: A scalable Random Early Detection scheme with flow trust against DoS attacks," IEEE Communications Letters, Vol. 17, No. 5, pp. 1032-1035, 2013.

[9] Saied, A., Overill, R. E., & Radzik, T, "Detection of known and unknown DDoS attacks using Artificial Neural Networks," Neurocomputing, Vol. 172, pp. 385-393, 2016.

[10] Sharma, R, "Detection of Low-Rate DoS Attacks against HTTP Servers using Spectral Analysis," 2014.

[11] Alomari, E., Manickam, S., Gupta, B. B., Karuppayah, S., & Alfaris, R, “Botnet-based distributed denial of service (DDoS) attacks on web servers: classification and art,” International Journal of Computer Applications, Vol. 49, No. 7, pp. 24-32, 2012.

[12] Tanha, M., Torshizi, S. D. S., & Shamala, S, “A discrete event simulator for extensive defense mechanism for denial of service attacks analysis,” American Journal of Applied Sciences, Vol. 9, No. 6, pp. 909, 2012.

[13] M. Sharma, N. Unde, K. Borude and A. Paradkar, "A data mining based approach towards detection of low rate DoS attack," International Conference for Convergence for Technology, Pune, pp. 1-6, 2014.