Two Phase Detection Process to Mitigate LRDDoS Attack in Cloud Computing Environment

Amrutha Muralidharan Nair[1], Dr. R Santhosh[2]
Research Scholar, Department of Computer Science Engineering, Karpagam Academy of Higher Education, Coimbatore, India[1]
Professor, Department of Computer Science Engineering, Karpagam Academy of Higher Education, Coimbatore, India[2]

***Abstract*—Distributed Denial of Service (DDoS) is a major attack carried out by attackers leveraging critical cloud computing technologies. DDoS attacks are carried out by flooding the victim servers with a massive volume of malicious traffic over a short period, Because of the enormous amount of malicious traffic, such assaults are easily detected. As a result, DDoS operations are increasingly appealing to attackers due to their stealth and low traffic rates, DDoS assaults with low traffic rates are also difficult to detect. In recent years, there has been a lot of focus on defense against low-rate DDoS attacks. This paper presents a two-phase detection technique for mitigating and reducing LRDDoS threats in a cloud environment. The proposed model includes two phases: one for calculating predicted packet size and entropy, and another for calculating the covariance vector. In this model, each cloud user accesses the cloud using the virtual machine, which has a unique session ID. This model identifies all LRDDoS assaults that take place by using different protocols (TCP, UDP, ICMP). The experiment's findings demonstrate, how the suggested data packet size, IP address, and flow behavior is used to identify attacks and prevent hostile users from using cloud services. The VM instances used by different users are controlled by this dynamic mitigation mechanism, which also upholds the cloud service quality. The results of the experiments reveal that the suggested method identifies LRDDoS attacks with excellent accuracy and scalability.**

Keywords—LRDDoS attack; distance deviation; covariance vector; threshold

I. Introduction

As next-generation Internet technologies are devised and developed, distributed denial of service (DDoS) attacks on the internet have become exceedingly dangerous. The traditional technique of executing DDoS attack is to flood the network with a high number of packets, straining the server's bandwidth, computational power, memory and delaying legitimate users' access to resources. In 2001, Asta networks observed a new sort of assault on the internet backbone: a denial-of-service attack. Kuzmanovic and Knightly discussed the idea at the SIG conference in 2003 [1]. They assumed the attacks were conventional DoS attacks, which may significantly limit and restrict network traffic. Furthermore, the attacks are called “Shrew Attacks" because they cannot be detected by the methodologies. The attacks were known as Low Rate Distributed Denial of Service (LRDDoS) attacks by other researchers. [2]. Unlike classic assaults, LRDDoS attacks send intermittent high-volume queries and use weakness in the network protocol [3] to actively minimize resource requirements for normal users. The attacks have a substantial impact on network performance.

An average attack traffic, on the other hand, is quite minimal due to the short duration of each assault burst, which is remarkably comparable to the burst traffic generated by many conventional application services. LRDDoS attacks are difficult to identify and mitigate because of their stealth and destructiveness. If the device requires new functionalities, the new protocols or regulations must be redesigned. As a result, existing network detection mechanisms for LRDDoS assaults are often down.

DDoS assaults are classified by the network into two sorts based on their behaviour: "Low Rate DDoS" and "High Rate DDoS" (LRDDoS & HRDDoS attacks) [4][17]. The HRDDoS attack's goal is to block legitimate users from accessing services. These attacks are carried out by transferring a large volume of traffic in order to take advantage of network capacity. The fundamental weakness of the HRDDoS assault is its traffic characteristics, which is why the attackers prefer the LRDDoS approach [5]. LRDDoS attacks are difficult to detect since the assault traffic resembles normal traffic.

Instead of depleting network bandwidth and resources throughput, an LRDDoS attack targets protocol stack flaws. The attacker emits malicious packets at a low rate, because of which the security systems built on network-level are not able to detect the characteristics of the attack. The attacker’s goal is basically degrading the "Quality-of-Service (QoS)" being experienced by a legal end user rather than disrupting the network services delivered to them. Many approaches to detecting DDoS attacks have been developed, including the Anomaly Detection System (ADS). However, LRDDoS attacks involve regular behavior as normal traffic deliberately, as a result of which, it is difficult to identify.

The goal of an LRDDoS attack is to continuously drain resources and bandwidth [4]. This form of assault generates adequate traffic in the network. Fig. 1 depicts the LRDDoS assault scenario. The network time duration (Δt), burst rate (br) and burst width (bw) are used to describe these assaults. LRDDoS attacks operate differently than traditional DDoS attacks. Since TCP vulnerabilities are the main targets of LRDDoS attacks, it can be difficult and complex to identify these attacks.

LRDDoS attacks differ significantly from traditional kinds of assault detected via anomaly detection techniques. This attack makes use of TCP congestion by transmitting malicious traffic in small bursts over a short period of time, known as a pulsing assault, or by sending packets at a steady pace, known as a constant attack. On average, present LRDDoS detection algorithms can identify just a small percentage of attack packets. Because the difference between regular traffic and LRDDoS traffic is so small, it is extremely difficult to recognize and discriminate.

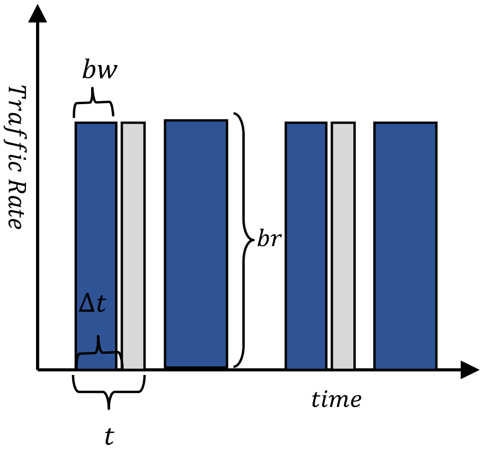


Fig. 1. LRDDoS attack traffic variations.

The present cloud security solutions are non-adaptive and insufficient for detecting LRDDoS attacks. To address this issue, a two-phase detection approach is used to distinguish between legitimate and malicious communication in the cloud computing environment. The suggested approach is independent of the assault pattern. It may achieve a significant distance barrier, leading to a low false positive rate.

The following is the contribution:

Examine the difference in packet size distribution between legitimate and malicious assaults.

A system that incorporates quick access to cloud services is recommended.

The suggested approach adapts to both internal and external network traffic, allowing for the detection of the attack and the reduction of LRDDoS recurrence in the future.

II. Related Work

Wang et al. [6] introduced METER, an “enseMble discrEte wavelet Transform-based technique for detecting low-frequency DDoS assaults in SDN”. This model assists in identifying the assault by computing the wavelet coefficients matrix and the associated entropy. Yu et al. [7], devised a methodology to identify DDoS attack using dynamic resource allocation approach and queueing theory. Xiang et al. [8] used information metrics such as generalized entropy and information distance to network traffic-based algorithms to detect low-rate DDoS attacks.

A mathematical model for recognizing low-rate DDoS assaults was developed by Luo et al. [9] based on the congestion characteristics of victim TCPs. Wu et al. [10] also developed a mathematical model that combines the MF-DFA algorithm with the holder exponent to distinguish between malicious and non-malicious traffic in a low-rate DDoS assault.

Takahashi et al. [11] developed a method for detecting a shrew DDoS assault that has already been initiated in a home network setting employing a bottleneck connection with unknown bandwidth and buffer capacity. The proposed attack detection method reduces downstream traffic from targeting network to keep the quality, while keeping the attack traffic covert by increasing the pulse rate exploratorily and measure the attack effect by deploying bot nodes in the home network.

By monitoring the pace at which flow table rules are applied, Dhawan et al. [12] suggested a technique for identifying DoS assaults. The network is alerted that it may be attacked when the rate of flow rule installation rises over a certain threshold, and the defensive mechanism is then turned on.

H. Chen et al. [13] offer a hybrid approach for detecting LDoS attacks that incorporates trust evaluation with the Hilbert-Huang Transformation. An intrinsic mode function (IMF) is implemented using a hybrid method which includes the correlation and Kolmogorov-Smirnov values, and if these values are more than 0.4 and 0.3, respectively, and the static point shows a higher degree of confidence in the network, it will help in detecting LDoS assaults.

Kieu et al. [14] suggested a technique for detecting LDDoS attacks by estimating TCP throughput and using the TCP congestion window. Wu et al. [15] identify and filter out DDoS traffic using temporal frequency analysis. The filtering technique is developed as a system in the real world.

Florea et al. [16] advocated adopting a unified detection architecture to overcome the challenge of detection against low-rate DDoS attacks.

III. System Modeling and Assumptions

The proposed model is a dynamic mitigation strategy for detecting LRDDoS attacks and optimizing QoS while working with constrained system resources. Both lawful packets and attack packets supplied by the legitimate user and the attacker may be easily sent to the current network detection system since the internet was created for openness and best effort transmission. In the event of an LRDDoS assault, when compared to lawful traffic, the attack packet shows several odd properties, such as the quantity of traffic flows and packets with unusual distributions or statistics [18]. LRDDoS packets have higher features than legitimate traffic since they are purposefully manufactured by prebuilt programme. As a result, the packet size in each request may be used to measure the distribution difference between regular traffic and malicious traffic [18]. While considering an LRDDoS attack, each and every packet transferred to the network is regarded a lawful request packet since all of the header information is acceptable. This helps the attacker to purposefully aggregate the packets and attack the victim’s server which leads to the display of abnormal deviations in its network [5].

The technique focuses on the differences between the distribution of packet sizes and between legitimate and attack packets. This measured difference allows for the identification of the traffic.

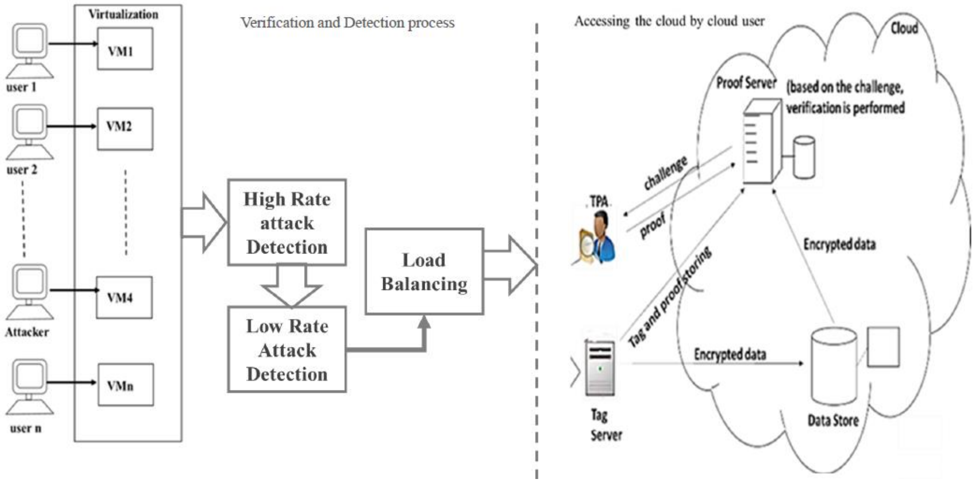


Fig. 2. Proposed model.

As seen in Fig. 2, each legal cloud user (cc_i) is given access to a virtual machine (VM_i). The cloud architecture is split into two portions, one to offer service and the other to verify and identify DDoS attacks. The cloud user's virtual machine $(VM_i)$is the source of the verification and detection process, where the LRDM algorithm is put and from which the predicted packet size of each traffic is calculated. The Low-rate detection method receives the network traffic from each virtual machine VM from which session ID S_{id_i} is retrieved.

A. PHASE I Algorithm

The LRDM method is split into two stages; the first stage examines the packet size and flags communication as malicious if it exceeds a certain threshold. Each user (cc_i) connects to the network and sends the (Rq_i) request packet to the cloud. Let $Rq_i(t)$ represent the collection of network flows in the cloud over the t interval.

$$Rq_i(t) = \{\, Rq_1(t), Rq_2(t), Rq_3(t), Rq_4(t), \dots \dots \dots \dots Rq_n(t)\}$$

The cloud user CC_i sends a request Rq_i with a packet size PS_i for a particular time interval Δt. The maximum packet size that can be sent across the network is set to $PS_{max} = 1514$. As a result, the network flow at the moment Δt for each cloud user will be $Rq_i(\Delta t)$.

$$Rq_i(\Delta t) = \{Rq_1(PS_1), Rq_2(PS_2), \dots \dots \dots \dots \dots Rq_n(PS_n)\,\}$$

It should be remembered that the network protocol limits packet size. As a result, TCP traffic during an LRDDoS assault will enter a malicious series of drop-recovery-drop. In LRDDoS attack situations, TCP transmission becomes more discontinuous and unsteady as compared to TCP traffic in normal network settings. The packet size received should satisfy the range $60 \leq PS_k \leq 1514$. Each cloud user sends a N_i network request to the cloud server at Δt time interval.

In order to create a collection of packet size arrays $PA(\Delta t)$, the packet size is directly retrieved from the packet header,

$$PA_i(\Delta t) = \{PA_1, PA_2, \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots PA_n\,\}$$

The mean packet size is calculated for the network over a time interval (Δt).

$$\overline{MPA} = \frac{1}{N}\sum_{k=1}^{N_i} PA_i\,, such\ that\ 60 \leq PA_k \leq 1514$$

The probability of occurrences of $Rq_i(\Delta t)$ is calculated as,

$$p\big(Rq_i(\Delta t)\big) = \frac{N_i}{\sum_{i=1}^{N} N_i}, where\ p\big(Rq_i(\Delta t)\big) \geq 0\ and \sum p\big(Rq_i(\Delta t)\big) = 1$$

The packet size expected for each network flow of CC_i is calculated as,

$$Ep'(x) = \sum_{i=1}^{N} p\big(Rq_i(\Delta t)\big) * \overline{MPA}$$

Next, compute the distance deviation between the calculated packet size $(Ep'(x))$ and default packet size $(Ep'_t(x))$, this distance gap help to identify the inequality between the legitimate traffic and normal traffic.

$$\partial(\propto, \Delta t) = Ep'(x) - Ep'_t(x)$$

Suppose $Rq = \{r'_1, r'_2, r'_3, r'_4\}$ is the ordered flow of traffic in the network at the sample time period Δt. Let the probability of each flow will be $P = \{p_1, p_2, p_3, p_4\}$. The mean packet size $\overline{MPA} = \{m_1, m_2, m_3, m_4\}$ for all value of Rq. The obtained and stored value of the normal expected packet size will show a loss of generality as,

$$Ep'\{r'_1, r'_2, r'_3, r'_4\,\} = Ep'\{r'_4, r'_3, r'_2, r'_1\}$$

The symmetry of the attack is independent of the arrival pattern and pulse pattern; therefore the accuracy will be overwhelmed by the distance deviation caused by the LRDDoS attack in the network.

$$Att_{flag} = \begin{cases} 1, & \partial(\propto, \Delta t) \geq 0 \\ 0, & \partial(\propto, \Delta t) < 0 \end{cases}$$

Fig. 3 depicts the flow diagram illustrating the Algorithm of phase 1.

```
Algorithm of Phase 1
Input:
Set of requests $Rq_i(\Delta t)$ and an array of packet size $ps_{max}$
Output:
Attack Detection or setting the packet with a flag $Att_{flag}$
Procedure VV ():
Retrieve the request's session id $S_{id_i}$
AnalysisLRDDoS ():
if $n_{rin} > 100$ then
Stop executing all requests with the same session ID $S_{id_i}$
else
Generate an array $PA_i(\Delta t)$
Calculate the mean packet size
    $\overline{MPA} = \frac{1}{N}\sum_{k=1}^{N_i} PA_i$
Calculate the probability occurrences $Rq_i(\Delta t)$,
    $p\big(Rq_i(\Delta t)\big) = \frac{N_i}{\sum_{i=1}^{N} N_i},$
Compute the expected packet size of the network for the cloud user $CC_i$,
    $Ep'(x) = \sum_{i=1}^{N} p\big(Rq_i(\Delta t)\big) * \overline{MPA}$
Distance deviation is calculated
$\partial(\propto, \Delta t) = Ep'(x) - Ep'_t(x)$
if $\partial(\propto, \Delta t) \geq 0$ $then,$
set the attack flag as $Att_{flag} = 1$ (i.e, LRDDoS atatck is detected)
else
set the attack flag as $Att_{flag} = 0$ (i.e, Normal traffic)
```

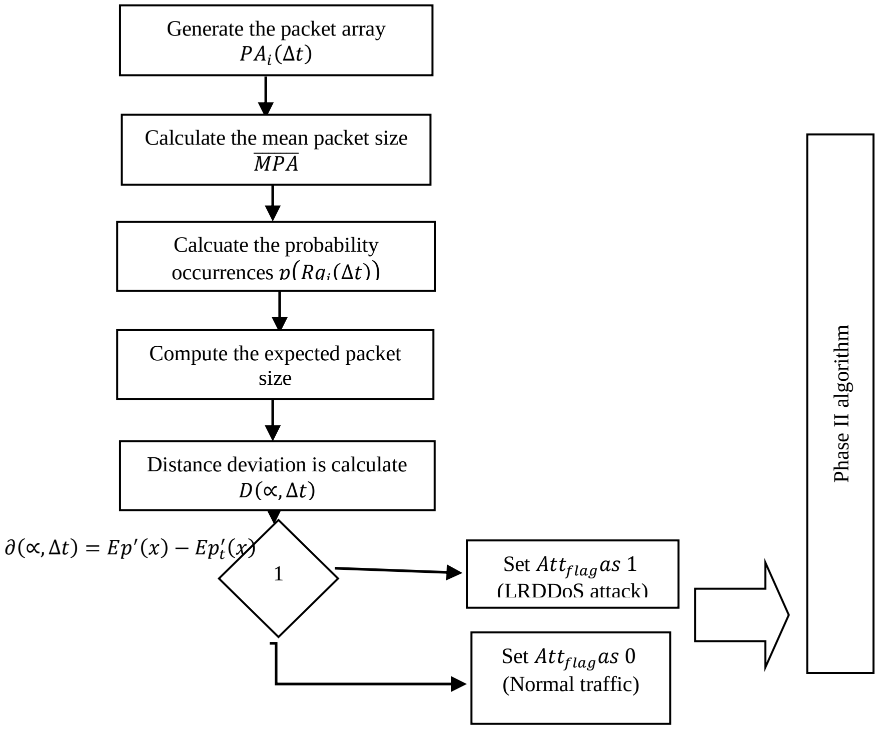

Generate the packet array $PA_i(\Delta t)$
Calculate the mean packet size $\overline{MPA}$
Calcuate the probability occurrences $p(Rq_i(\Delta t))$
Compute the expected packet size
Distance deviation is calculate $D(\propto, \Delta t)$
$\partial(\propto, \Delta t) = Ep'(x) - Ep'_t(x)$
1
Set $Att_{flag} as$ 1 (LRDDoS attack)
Set $Att_{flag} as$ 0 (Normal traffic)
Phase II algorithm

Fig. 3. Phase I flow algorithm.

B. PHASE II Algorithm

In this phase II process, a usage of covariance vector to identify LRDDoS assaults; if the LRDDoS assault pulse is perceived as a significant signal, the network background traffic acts as the sender's noise. During transmission, attack flows are masked by genuine traffic; nevertheless, covariance vector detection is used to identify attack flows at the receiving end.

1) The covariance vector principle: A random vector's covariance matrix is a square matrix that contains all of the covariances between the vector's entries. Consider the two vectors $\hat{x}$ and $\hat{y}$ which are used as random vector as,

$$\hat{x} = \{\, x_1, x_2, \dots\dots\dots\dots, x_n\}^t$$

$$\hat{y} = \{\, y_1, y_2, \dots\dots\dots\dots, y_n\}^t$$

$$cov[\hat{x},\hat{y}] = E[(x_i - E[x_i'])(y_j - E[y_j'])]$$

$cov[\hat{x},\hat{y}]$ is known as cross covariance vector. A cross-covariance matrix is one in which the element at the i, j positions represent the covariance between the i^{th}element of one random vector and the j^{th} element of another random vector variable with several dimensions. All of the scalar random variables in the vector are its elements. There is finite or an infinite number of potential values for each element, as well as a finite or an unlimited number of values that may be experimentally observed. The cross-covariance matrix logically adds more dimensions to the idea of covariance. Typically, the cross-variance vector mean of the two vectors x and y is expressed as,

$$K_{xy} = cov[\hat{x},\hat{y}] \overset{\text{def}}{=} Ep\big[[x_i - Ep(x_i')] - [y_i - Ep(y_i')]\big]^{\Delta t}$$

In the phase II algorithm, some initial vectors are used to perform the calculations for each request coming from the cloud user CC_i. A normal traffic vector is generated $\widehat{R_{nor}} = \{RN_1, RN_2, \dots\dots\dots . RN_n\}$, mean vector value of the normal traffic as M_{nor} and threshold value δ_{nor}. In this phase the covariance vector mean value is subtracted by predefined mean vector value and then compared with the 3-phase threshold and 4-phase threshold value to detect the LRDDoS attack in depth. The request (Rq_i) which is detected will be put on hold for a specific time period and the IP address (IP), protocol used(P), attack duration (D_l), attack period(Δt), attacking rate(r) is blacklisted which is indicated as B_{list}. Fig. 4 depicts the flow diagram illustrating the Algorithm of phase II.

```
Algorithm of Phase II
Input :
Set of request $R_i(\Delta t)$
$\widehat{R_{nor}}$ mean vector value of the normal traffic as $M_{nor}$
Threshold value $\delta_{nor}$.
Output:
$B_{list}$ and time of halt
Procedure :
Convert the request obtained from the network to a vector form $\widehat{R_{rec}}$
Check the result obtained from the from the pahse I algorithm
İf $Att_{flag} = 1$
Blacklist the request $R_i(\Delta t)$ in $B_{list}$ file by capturing the information as IP address $(IP)$,Protocol used$(P)$,attack duration $(D_l)$, attack period $(\Delta t)$, attack rate $(r)$and halt the cloud user for $\frac{1}{2^n}\Delta t$.
Otherwise
Compute the covariance vector mean as,
    $K_{rec} = E\big[[R_{inor} - E(R'_{inor})] - [R_{irec} - E(R'_{irec})]\big]^{\Delta t}$
İf $K_{rec} - M_{nor} \le 3\delta_{nor}\ or\ 4\delta_{nor}$ then,
Set $Att\_flag$=0, consider as normal traffic
Else
Set $Att_{flag} = 1$ and Blacklist the request $R_i(\Delta t)$ in $B_{list}$ file by capturing the information as IP address $(IP)$,Protocol used$(P)$,attack duration $(D)$, attack period $(\Delta t)$, attack rate $(r)$. and halt the cloud user for $\frac{1}{2^n}\Delta t$
```

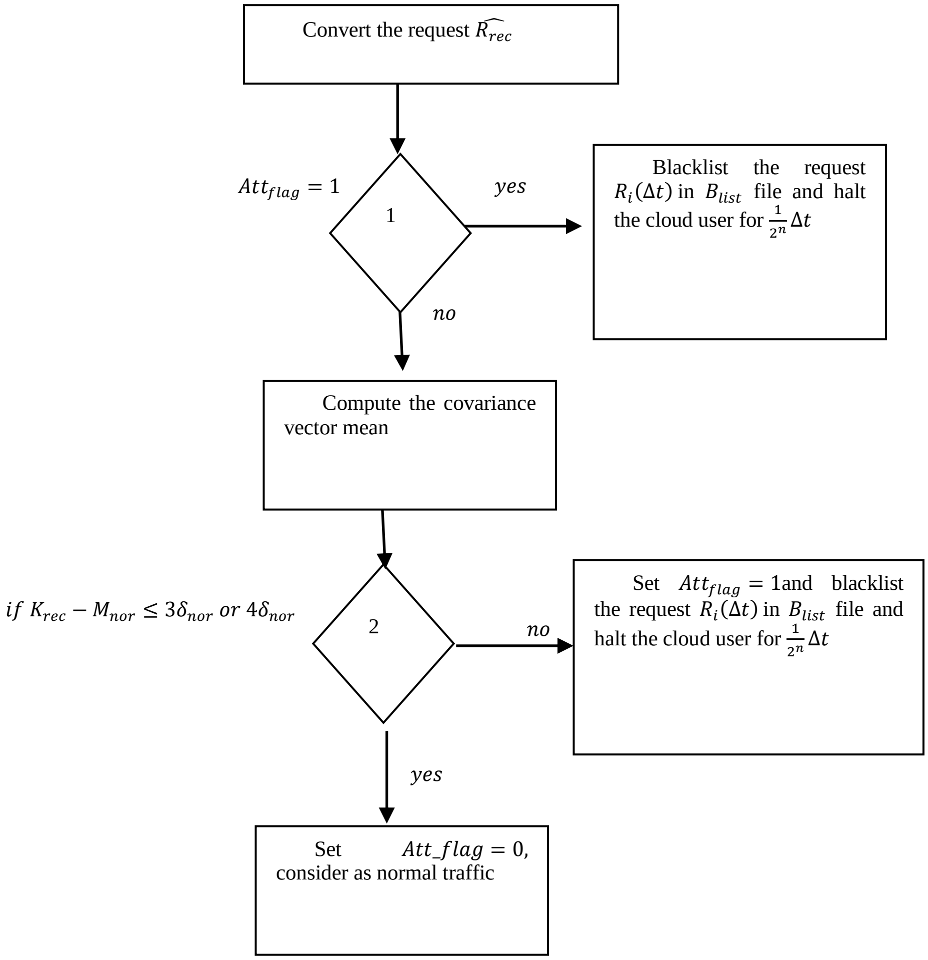

Convert the request $\widehat{R_{rec}}$
$Att_{flag} = 1$
1
yes
Blacklist the request $R_i(\Delta t)$ in B_{list} file and halt the cloud user for $\frac{1}{2^n}\Delta t$
no
Compute the covariance vector mean
$if\ K_{rec} - M_{nor} \leq 3\delta_{nor}\ or\ 4\delta_{nor}$
2
no
Set $Att_{flag} = 1$and blacklist the request $R_i(\Delta t)$ in B_{list} file and halt the cloud user for $\frac{1}{2^n}\Delta t$
yes
Set $Att\_flag = 0$, consider as normal traffic

Fig. 4. Phase II flow algorithm.

IV. Experimental Outcome and Analysis

The experiment is made by establishing some crucial criteria depending on the outcomes of available resources in order to assess how well the suggested system performs when subjected to an LRDDoS attack, identify the attack's features, and creates a blacklist of the attacker. The primary justification for looking into the assault is the design of the communication protocols, which enable successful end-to-end service delivery. The proposed system mitigates LRDDoS attacks in a cloud computing environment using a two-phase detection method. Attackers however, use the protocols to change how cloud services and apps are accessible. According to the suggested strategy, the three main flooding assaults are TCP SYN, UDP, and ICMP. In this case, the victim end of the attack experiences one-way attack traffic and disproportionately high levels of unauthorized resource use.

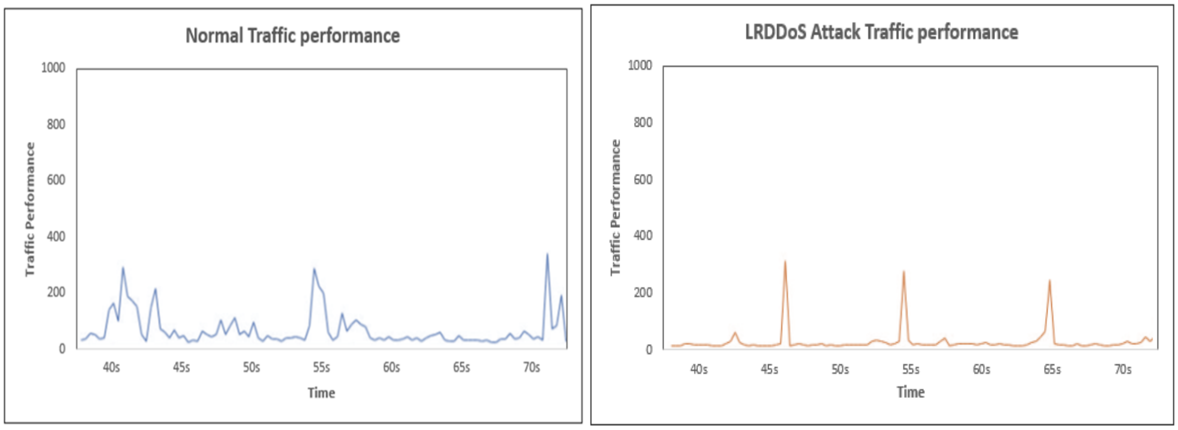
The experimental setup uses the following configuration. One server with $m + n$ virtual machines is used, where m stands for legitimate users and n stands for malicious users. The configuration includes Windows 10 operating system, Intel (R) Core (TM) i7-4790 CPU running at 3.60 GHz, 500 GB of storage, 16 GB of RAM for desktops and 72 GB for server. The scenarios of low-rate DDoS attacks and non-attack traffic are created for the experiment. In the absence of an attack, authorized users (m virtual system) contact the cloud server to request access to services or files kept on the server. A malicious user (n virtual machine) sends attack packets to the cloud server in an attack scenario. In contrast, in an attack scenario, the cloud server will simultaneously receive requests from both legitimate users and malevolent users. To identify the LRDDoS attack sources per source IP throughout the observed period, the correlation 1 query is run on the input stream. By combining the source addresses that come from the user group ($m + n$ users), correlation is achieved. At the selected time interval, network flows are gathered. Attack packets are added to the Backlist B_{list} along with the session ID of the virtual machine source when the individual source IP crosses the threshold value.

The two-phase detection system captures the stream processing quite well. Data from the baseline profile are used to calculate the threshold value. The cloud's statistics and behavior while it is not under assault are examined. The baseline profiling is done for a period of time that varies from daily to weekly to monthly. The baseline is periodically updated to reflect any alterations to the cloud usage. The Slowloris attack tool is used to simulate low-rate attack situations. It is a DDoS attack programme that creates irregular HTTP connections using the Hyper Text Transfer Protocol (HTTP). By absorbing all connections of the server, the tool attempts to maintain HTTP connections for an extended period of time while slowing down the cloud server (such as Apache and dhttpd). The Apache web server's timeout setting is 250 seconds by default; however, it can be changed depending on how the attack packets are transmitted.

Incomplete HTTP connections are opened by the Slowloris to launch low-volume assaults. It performs data requests and, once all connections have been used, resets the timeout counter value to 1. Using $x + y$ malicious nodes, the assault begins with the command perl $slowloris.pl. -dns - s\ 192.169.60.1 - 250\ www.abcexample.com\ - timeout\ 3\ - num\ 60\ - port\ 80$. By claiming to be IP addresses in the subnet from 192.169.60.1 to 192.169.60.250, the malicious nodes create 60 connections and maintain them by making data requests every 5 seconds. The low acquisition rate timeout of 5 seconds is selected. In low-rate attack scenarios, the legitimate nodes submit request packets to the cloud server concurrently with the malicious nodes. At the CC, the traffic is recorded for each scenario, and a matching Traffic flow behavior graph (TFBG) is created. In Fig. 5, the TFBGs are shown. Fig. 5(a) demonstrates the steady traffic flow that was present when there was no threat of assault, while Fig. 5(b) depicts the periodic and pulsating traffic streams seen during an LRDDoS assault.

The detection precision of each assault is displayed in Table I below. The total detection accuracy for a TCP SYN flooding attack is stated to be 99.97% in a time window of 60 seconds. For different threshold values, such as 100, 1000, 5000, 10,000, and 15,000, the accuracy of the attack detection is accordingly 99.85%, 99.98%, 99.99%, 99.99%, and 99.99%. The overall detection accuracy for a UDP flooding attack is reported to be 99.96% during a time window of 60 seconds. Attack detection accuracy ranges from 99.81%, 99.98%, 99.99%, 99.99%, and 99.99% for threshold values of 100, 1000, 5000, 10,000 and 15,000, respectively. For an ICMP flooding assault, the total detection accuracy was 99.84% within a 60-second time window. The accuracy with which the assault is detected is 99.32%, 99.93%, 99.98%, 99.99% and 99.99% for various threshold values, such as 100, 1000, 5000, 10,000 and 15,000, respectively.

In accordance with the IP addresses, the received packets are counted. The non-attack scenario had an average flow count of 2982; the low-rate assault scenario had an average flow count of 610. Table II provides an overview of the values for the selected parameters. According to Fig. 6, legitimate requests in the system under the nonattack scenario last a little bit longer than malicious ones under the LRDDoS attack scenario. The resource isolation means that the LRDDoS assault won't have an effect on the container instances handling malicious requests that have been blacklisted. Fig. 6 demonstrates the flow of requests within the network, showcasing both normal traffic and malicious traffic.



(a) (b)

Fig. 5. Traffic flow behavior graph: (a) Normal traffic, (b) LRDDoS attack traffic.

TABLE I. LRDDOS ATTACK DETECTION PRECISION

Attack Source	Threshold	TCP SYN	UDP	ICMP
SET 1	100	99.85	99.81	99.32
	1000	99.98	99.98	99.93
	5000	99.99	99.99	99.98
	10000	99.99	99.99	99.99
	15000	99.99	99.99	99.99
Average		99.97%	99.96%.	99.84%

TABLE II. PARAMETER EXPERIMENTAL VALUES

<table>
<tr><th colspan="3">Traffic scenario</th><th>Behavior graph pattern</th><th>Average flow count</th><th>Response time</th></tr>
<tr><td colspan="3">Normal Traffic</td><td>Uniform pattern</td><td>2982</td><td>60 seconds</td></tr>
<tr><td>LRDDoS attack traffic</td><td>Periodic and pulsing</td><td>610</td><td colspan="3">60 seconds</td></tr>
</table>

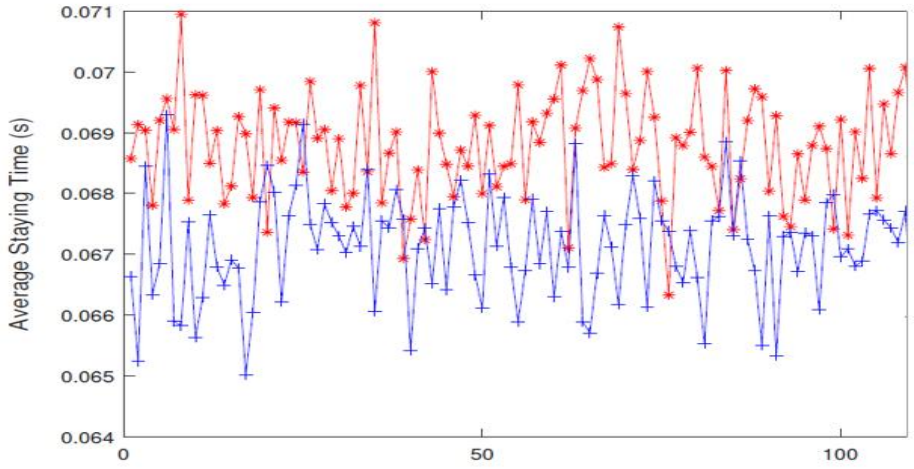


Fig. 6. Combined traffic flow behavior.

Table III provides a comparison of the suggested technique to the current approaches. When comparing approaches, it is taken into account how well they can identify DDoS attacks with low attack rates, as well as how quickly they can respond. To evaluate the correctness of the proposed technique, two metrics are used: True Negative Rate (TNR) and True Positive Rate (TPR), also known as specificity and sensitivity, respectively. TPR and TNR are calculated as follows:

$$TPR = \frac{x'}{x' + y'}$$

and

$$TNR = \frac{a'}{a' + b'}$$

where,

$x' =$ *signifies properly identifying malicious users.*

$y' =$ *Unauthorized users .*

$a' =$ *Accurately detected authorised users.*

$b' =$ *Authorized users who were mistakenly labelled malevolent.*

The following metrics are used to assess the proposed method's accuracy:

$$Accuracy = \frac{x' + a'}{x' + y' + a' + b'}$$

The experiment was repeated multiple times with various combinations of reliable $(x + p)$ and malicious $(y + q)$ nodes selected from the set of 16, 32, 64, 128, 256, and 512 in order to evaluate performance. Table III displays the comparative analysis of the experimental outcomes. For a high number of nodes, there is only a little change in the flow count. Therefore, only the results for up to 128 nodes are displayed. Based on the response time, average flow count, and traffic flow behavior, the study is carried out. Based on the trials, 99.1% and 99.5%, respectively, are the average values for TPR and TNRs shown in Fig. 7. The proposed technique has a 99.4% total accuracy rate as shown in the below Table III.

TABLE III. COMPARISON OF PROPOSED SYSTEM WITH EXISTING SYSTEM

Methodology	Normal	LowRate	Response time	Accuracy
FR-Red [a]	Yes	Yes	(15.1-125.8) sec	98%
Queue based model[b]	Yes	Yes	Medium	99.1%
LORD [c]	Yes	Yes	75sec	99.1 %
MPTCP [d]	Yes	Yes	33 sec	98.9%
Proposed Methodology	Yes	Yes	(60-65) sec	99.4%

**NA= Not Available*

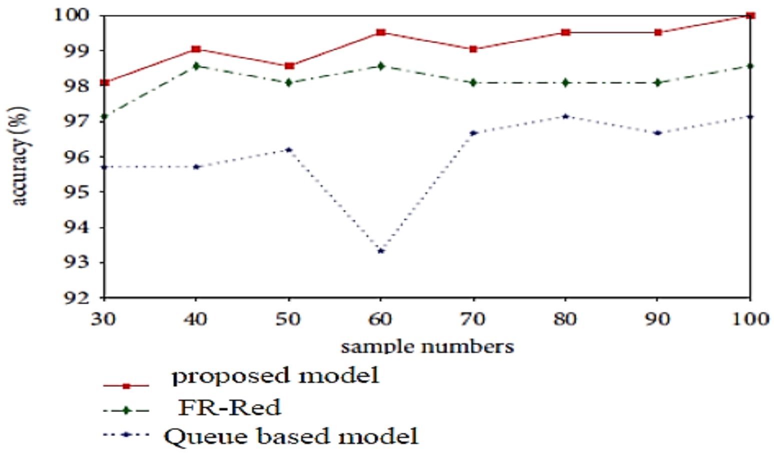


Fig. 7. A comparison between the proposed methodology and existing methodology.

V. Conclusion and Future Research Prospects

This work proposes a simple and effective two-phase detection methodology for detecting LRDDoS attacks in a cloud domain. The suggested model incorporates the 3D Threshold and covariance vector notions as well as the packet size of each request and the network flow count. Based on the concept, a faulty LRDDoS attack in a network with restricted resources is possible. The model's performance demonstrates a 99.97%, 99.96%, and 99.84% accuracy in identifying the LRDDoS attack, which is carried out utilizing the TCP, UDP, and ICMP protocol. By establishing a variety of threshold values, the testing results show an average detection of 99.8%. Using this method, the cloud user’s virtual machine may be optimized and system resources can be dynamically reassigned. The suggested approach entirely eliminates the effect of LRDDoS attack by blacklisting the user for a particular time period and increase the ability of the other user to utilize the service of cloud without get affected by the assault.

As part of future work, it is necessary to investigate strategies to counteract LRDDoS attacks on the assumption that virtual machines may expand with limitless resources. Furthermore, in the limitless resources scenario, an attempt is made to investigate pricing difficulties in a VM-based cloud system when defending against an LRDDoS assault.

References

[1] Kuzmanovic, E.W. Knightly, Low-rate TCP-targeted denial of service attacks: the shrew vs. the mice and elephants, in: Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, 2003, pp. 75–86.

[2] G. Maciafernandez, J.E. Diazverdejo, P. Garciateodoro, Evaluation of a low-rate DoS attack against iterative servers, Comput. Netw. 51 (4) (2007) 1013–1030.

[3] J. Nagle, Congestion control in IP/TCP internetworks, ACM SIGCOMM Comput. Commun. Rev. 14 (4) (1984) 11–17.

[4] V.Adat, A.Dahiya, and B.Gupta, “Economic incentive based solution against DDoS for IOT customers,” in ICCE. IEEE, 2018, pp. 1-5

[5] J. Idziorek, M. F. Tannian, and D. Jacobson, “The insecurity of cloud utility models,” IT Professional, vol. 15, no. 2, pp. 22–27, 2013.

[6] C. Wang, Y. Cui, Q. Qian, G. Shen, H. Gao and S. Li, "METER: An Ensemble DWT-based Method for Identifying Low-rate DDoS Attack in SDN," 2021 IEEE 19th International Conference on Embedded and Ubiquitous Computing (EUC), 2021, pp. 79-86, doi: 10.1109/EUC53437.2021.00020.

[7] S. Yu, Y. Tian, S. Guo, and D. O. Wu, “Can we beat DDoS attacks in clouds?” IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 9, pp. 2245–2254, 2014.

[8] Y. Xiang, K. Li, and W. Zhou, “Low-Rate DDoS Attacks Detection and Traceback by Using New Information Metrics,” IEEE Transactions on Information Forensics and Security, vol. 6, no. 2, pp. 426–437, 2011.

[9] J. Luo, X. Yang, J. Wang, J. Xu, J. Sun, and K. Long, “On a mathematical model for low-rate shrew DDoS.” IEEE Transactions on Information Forensics and Security, vol. 9, no. 7, pp. 1069–1083, 2014.

[10] Z. Wu, L. Zhang, and M. Yue, “Low-rate dos attacks detection based on network multifractal,” IEEE Transactions on Dependable and Secure Computing, vol. 13, no. 5, pp. 559–567, 2016.

[11] Y. Takahashi, H. Inamura and Y. Nakamura, "A Low-rate DDoS Strategy for Unknown Bottleneck Link Characteristics," 2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops), pp. 508-513,2021

[12] M. Dhawan, R. Poddar and K. Mahajan, "SPHINX: Detecting security attacks in software-defined networks", Proc. Netw. Distrib. Syst. Secur. Symp., pp. 8-11, Feb. 2015.

[13] Rate Denial of Service Attack Detection Approach in ZigBee Wireless Sensor Network by Combining Hilbert-Huang Transformation and Trust Evaluation,” IEEE Access, vol. 7, pp. 32 853–32 866, 2019.

[14] M. V. Kieu, D. T. Nguyen and T. T. Nguyen, "A Way to Estimate TCP Throughput under Low-Rate DDoS Attacks: One TCP Flow," 2020 RIVF International Conference on Computing and Communication Technologies (RIVF), 2020, pp. 1-8

[15] Z. Wu, W. Cui and P. Gao, "Filtration method of DDoS attacks based on time-frequency analysis," 2021 7th IEEE Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS), 2021, pp. 75-80

[16] R. Florea and M. Craus, "Modeling an Enterprise Environment for Testing Openstack Cloud Platform against Low-Rate DDoS Attacks," 2022 26th International Conference on System Theory, Control and Computing (ICSTCC), 2022, pp. 146-151

[17] N. Agrawal and S. Tapaswi, "A Lightweight Approach to Detect the Low/High Rate IP Spoofed Cloud DDoS Attacks," 2017 IEEE 7th International Symposium on Cloud and Service Computing (SC2), pp. 118-123, 2017.

[18] Lu Zhou, Mingchao Liao, Cao Yuan, Haoyu Zhang, "Low-Rate DDoS Attack Detection Using Expectation of Packet Size", Security and Communication Networks, vol. 2017, Article ID 3691629, 14 pages, 2017.