ISSN:2320-0790

An International Journal of Advanced Computer Technology

# Enhanced RRED to Counter LDoS Attacks

[1]Poornima.K.S, [2]Nandhni.J.P

[1,2]Department of Information Technology, SSN College of engineering, kalavakkam, PIN-603110
[1]k.pooja.222@gmail.com
[2]nandhinijprasad@gmail.com

***Abstract***: This paper provides an enhancement technique and algorithm to the Robust RED packet dropping algorithm in order to intelligently make packet drops and also to trace the IP addresses on a rough approximate of the creators of the low rate DoS attack. This technique also reduces the time complexity of the algorithm.

*Keywords*— RED, DoS, low rate, robust

## I. INTRODUCTION

As per current estimates, almost 41.1% [1] of the internet users have experienced security problems. Amongst all the attacking methods, one of the legacy methods-DoS (contributing to almost 16.8% [1] of the total kinds of attacks) still remains to be under existence. DoS or denial of service simply means flooding of packets from a large number of attackers to render a server useless for a particular amount of time. DoS can be created by SYN flooding, over flooding the victim with ping packets and teardrop attacks. Also a low rate DoS [2] attack which is used these days is an undetectable way of reducing TCP throughput. It exploits TCP's retransmission timeout mechanism to reduce TCP throughput without being detected. Compared to traditional flooding based DoS attacks, the low-rate DoS attack does not employ a "sledge-hammer" approach of high-rate transmission of packets, and consequently eludes detection. RED-like algorithms have already been found to be notably vulnerable to LDoS [3] attacks.

This paper provides an improvisation algorithm to the RRED congestion control algorithm[2] of TCP which filters packets and also to spot the IP addresses of low rate DoS attackers roughly more quickly based on an extra IP address comparison. Based on the frequency of the hardwares and the type of attack , the application front end hardware chosen here is a router. The router is the bottleneck and this algorithm is implemented by the router.

## II. RED

The Random Early Detection Algorithm (RED) had been proposed to be used in the implementation of AQM(explained in Section 1). For each packet arrival the average queue size, *qn*, is calculated using the Exponential Weighted Moving Average (EWMA) . The average queue size so computed is compared with the minimum threshold *(minth*) and the maximum threshold *(maxth*) to determine the next action. The basic RED algorithm can be summarized as follows: If the $qn \leq minth$, then noincoming packets are marked or dropped. If $minth \leq qn \leq maxth$, then the arriving packet is marked/dropped with probability *pb*, which is given by: $pb \leftarrow maxp(qn - minth)/(maxth - minth)$. Finally, if we have $qn > maxth$ then all incoming packets are marked/dropped. To make the inter-packet drop uniform instead of geometric [4] suggests to use, $pa \leftarrow pb/(1 - count \cdot pb)$ as the marking/dropping probability, where *count* indicates the number of packets forwarded since last mark/drop. Recent studies have shown that RED is vulnerable to LDos attacks.[5]

## III. LDOS ATTACKS

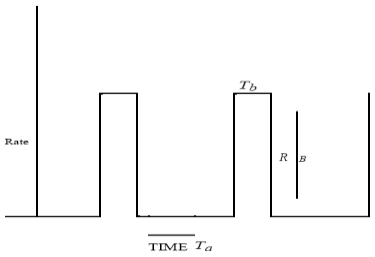


Fig 1: LDoS attack Stream

Following the notations in figure 1, we describe an LDoS attack using three parameters ( Ta,Tb,Rb ). As shown in Fig. 1, represents the attack period, represents the attack burst width, and represents the attack burst rate.

The LDoS attack exploits TCP's slow-time-scale dynamics of retransmission time-out (RTO) mechanisms to reduce TCP throughput [3]. Basically, an attacker can cause a TCP flow to repeatedly enter a RTO state by sending high-rate (Ta ), but short-duration bursts (Tb), and repeating periodically at slower RTO time-scales (Rb ). The TCP throughput at the attacked node will be significantly reduced while the attacker will have low average rate making it difficult to be detected. A critical observation needs to be noted here. Within a benign TCP flow, the sender will delay sending new packets if loss is detected (e.g., a packet is dropped). Consequently, a packet is suspected to be an attacking packet if it is sent within a short-range after a packet is dropped. This is the basic idea of our detection algorithm presented in Section IV.

## IV. ROBUST RED (RRED)

In this section, we explain the design and implementation of the RRED algorithm [7]. Fig. 2 describes the basic architecture of the RRED algorithm. A detection and filter block is added in front of a regular RED [4] block on a router. The basic idea behind the RRED is to detect and filter out LDoS attack packets from incoming flows before they feed to the RED algorithm. How to distinguish an attacking packet from normal TCP packets is critical in the RRED design. This is achieved base on the observation mentioned in Section III.

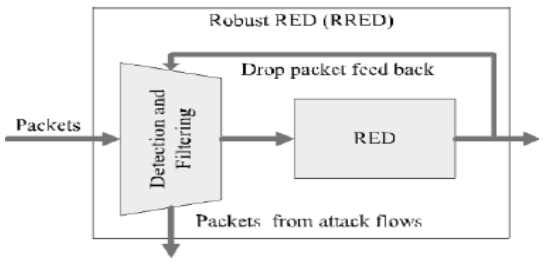


Fig. 2. Architecture of Robust RED (RRED)

All incoming TCP packets to a router belong to different flows. Here, a flow is defined by a 5-tuple *(Source IP, Source Port, Destination IP, Destination Port, Protocol)*. We use an indicator f . to judge whether flow is an LDoS attack flow or a normal TCP flow. Specifically, f. is calculated as follows. If a packet from flow is considered to be an attacking packet (described below),f .*I* is decreased by one; if it is considered to be a normal packet, .*I* is increased by one. Then an incoming packet from a flow with a negative I . is filtered. Packets from a flow with a positive or zero f will further feed to the RED block.

## V. ENHANCED RRED

The RRED[7] has a disadvantage with regard to dropping of packets. Packets are simply dropped based on the parameter of flow f at the end stage of the algorithm. No mechanism of storage exists to identify the IP of the attacker within immediate times of the DoS attack.

The enhanced RRED [7] is a novel approach that uses hashing of IP addresses of the most frequent packet sender and drops only those packets accordingly in addition to this RRED detection of DoS First of all the occurrence of a DoS is detected based on the frequency of the dropping of packets using T1 and T2 times.

An incoming packet from flow $f$ is suspected to be an attacking packet if it arrives within a short-range after a packet from $f$ that is dropped by the detection and filter block or after a packet from any flow that is dropped by the RED block. The following process is used to define this short-range. For every flow $f$ (either a normal TCP flow or an LDoS flow), let $f.T1$ be the arrival time of the last packet from $f$ that is dropped by the detection and filter block. Let $T2$ be the arrival time of the last packet from any flow that is dropped by the RED block.The short-range is defined as $[Tmax, Tmax + T^*]$, in which $Tmax = MAX(f.T1, T2)$. If the arrival time of an incoming packet from flow $f$ falls into this range, the packet is suspected to be an attacking packet. Note that $T1$ is flow specific while $T2$ is global, which capture the fundamental characteristics of an LDoS attack flow and the global impact of the attack on the whole network, respectively.

A proper value should be chosen for $T^*$ to (i) filter most attacking packets, and to (ii) pass most normal packets. In this letter, we empirically choose $T^*$ to be 10ms, which works quite well for diverse LDoS attacks.

$RRED - ENQUE(pkt)$

1: $f \leftarrow RRED - FLOWHASH(pk\text{t})$ and $FLOWHASH(pk\text{t})$ not flagged
2: $T_{max} \leftarrow MAX(T_1, T_2)$
3: if $pkt.arrivaltime \in [T_{max}, T_{max} + T_*]$ then
4: flag Flowhash(pkt)
5: else
6: $RED{-}ENQUE(pkt)$ $//pass\ pkt\ to\ the\ RED\ block$
7:end if
8: if $RED\ drops\ pkt$ then
9: $T_2 \leftarrow pkt.arrivaltime$
10: end if
11: else if flowhash[pkt] is set
12: $T_1 \leftarrow pkt.arrivaltime$
13: $drop(pkt)$
14: end if
15: return

Fig. 3. Pseudo code of the enhanced RRED algorithm

Fig. 3 shows the pseudo codes of the RRED algorithm. In Fig. 3, $pkt$ denotes an incoming packet; $f$ is the flow index hashed using$pkt$'s source-destination address pair via function $RRED -FLOWHASH()$. This flowhash function hashes the packets with the IP address of the packet as the key into a structure where the size if table is 255, since the maximum number of devices that can be connected to a router is 255.

The Flowhash structure contains a flag field and another field to record the incoming times of the packets. T1 is used to store the times of the packet that were recently dropped by the RED block.

## VI. WORKING

The packets on arrival are first hashed based on their IP addresses using a suitable hash function. After the hash function is applied, it is checked if that packet was flagged earlier or not. If it was flagged earlier it is simply discarded and the algorithm exits for that packet. If the packet was not flagged, if it checked if the packet was arrived in that given time interval ; in this case 10 ms (time interval between the recently dropped packet by the RED block or the detection block). If yes, then the IP is flagged and the packet is dropped by the detection block itself else the packet is forwarded to the RED block

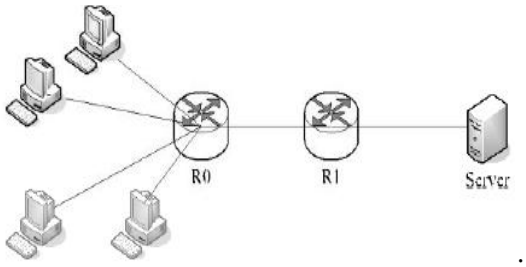


Fig. 4: Ro is Bottleneck

## VII. TESTING

On partial implementation of the algorithm at the filtering and detection block in java using multithreading and the required LDOS attack conditions set to mimic the real time attack with two specific cases; in accordance to the real time LDoS scenarios significant results were observed.

A system of two nodes, an attacker which mimics many attackers transmitting at the same time (by fixing high rates of TCP transmission ) and a user with a comparatively lower data rate was chosen.
The user was assumed to transmit 70 packets with the given delays while the attacker was assumed to transmit 2000 packets with the given delays(worst case scenario that corresponds to many attackers transmitting to the router at the same time).

The first set of percentages in fig.5 corresponds to the attackers transmitting at the rate of 1 packet per 1ms and the second sets of percentages correspond to the attackers transmitting at the rate of 1 packet per 5ms.

## VIII. CONCLUSIONS

The proposed algorithm increases the efficiency of the RRED detection and filtering block thus increasing the TCP throughput. Furthermore the drop rate at the RED block would be reduced since the load to the block is partly filtered and scanned before queuing to the RRED queue.

The percentages correspond to the number of packets transmitted out of the detection block after the application of the algorithm.

| 1 packet per 200 ms | | 1 packet per 100ms | | 1 packet per 50ms | |
|---|---|---|---|---|---|
| User | Attacker | User | Attacker | User | Attacker |
| 100% | 0.55% | 100% | 1.4% | 100% | 2.05% |
| 100% | 2.7% | 100% | 3.4% | 100% | 4.45% |

Fig. 5. Results of partial testing of RRED detection and filtering block

## IX. ADVANTAGES

The LDoS attacks that are difficult to detect can be detected using this algorithm. With the hash structure consisting of a flagging mechanism, the attacker IP addresses can be detected and the packet filtering also can be done. Since the algorithm involves comparing if the IP is flagged or not at the early stages of RRED technique, the time complexity with respect to comparing and then finding the attacker packet is reduced since a simple search is enough rather than performing the decrement/increment operation to the indicator bins. Also a sufficient intimation mechanism can be attached to the router to the nearest other monitoring device in order to notify other routers that a DoS attack has taken place nearby as an extension to this algorithm.

## X. FURTHER WORK

The algorithm is being improved to be implemented on a NS2 simulator. The main focus of work in future is to have a dynamic T* which changes on the fly as the network traffic conditions and better filtering of attacker packets so as to reduce the attacker passed AY 2010 on to the RED block as low as possible which would reduce the oscillations in the queue size of the RED block and also maintaining a constant TCP throughput in all traffic condtions.

## REFERENCES

[1] 15th Annual Computer Crime and cyber security survey (2010/2011)

[2] http://en.wikipedia.org/wiki/Denial-of-service_attack

[3] A. Kuzmanovic and E. W. Knightly, "Low-rate TCP-targeted denial of service attacks and counter strategies," *IEEE/ACM Trans. Netw.*,vol.14, no. 4, pp. 683–696, 2006R. E. Sorace, V. S. Reinhardt, and S. A. Vaughn, "High-speed digital-to-RF converter," U.S. Patent 5 668 842, Sept. 16, 1997.

[4] Congestion Control Algorithms in High Speed Telecommunication Networks.Aun Haider, Harsha Sirisena, Krzysztof Pawlikowski and Michael J.Ferguson ,http://www.orsnz.org.nz/conf36/papers/Haider.pdf

[5] M. Guirguis, A. Bestavros, and I. Matta, "Exploiting the transients of adaptation for RoQ attacks on Internet resources," in IEEE ICNP, 2004.

[6] RRED: Robust RED Algorithm to Counter Low-Rate Denial-of-Service Attacks Changwang Zhang, Jianping Yin, Zhiping Cai, and Weifeng Chen, IEEE COMMUNICATIONS LETTERS, VOL. 14, NO. 5, M