AN EFFICIENT DETECTION AND COUNTERMEASURE OF A JELLYFISH ATTACK USING DMPD ALGORITHM IN MANET

E. Sam Prabhakar*1 and Mr. K. Srinivasan*2

*PG Student, Department of Computer Science and Engineering, Sri Venkateswara College of Engineering, Sriperumbudur, India

Assistant professor, Department of Computer Science and Engineering, Sri Venkateswara College of Engineering, Sriperumbudur, India

Abstract— Critical applications to be deployed over Mobile ad hoc network (MANET) need security as a key factor. Insecure wireless communication and multi hop routing communication process leads to various types of attacks, by which the MANETs become vulnerable. In our project we analyze the behavior and impact of JellyFish attack over the TCP-based MANETs. The three types of JellyFish attack are JF-reorder, JF-delay and JF-drop. These attacks disturb the communication process of closed loop protocols such as TCP, without disobeying the protocol rules. This leads to degradation in throughput. We proposed a new algorithm called direct multiple parameter-based detection algorithm (DMPD) to detect and remove a JellyFish node from an active communication route. In our proposed DMPD algorithm each node uses the various parameters which are collected a time period to identify whether its neighbor node is a JF-attacker or not. To provide more security the DMPD algorithm represents the parameters such as number of packets generated and forwarded by the neighboring nodes, the past activity of the node, signal strength, delay of forwarding packets, and throughput. The extensive simulation results are obtained using a network simulator (ns-2).

Index Terms— jellyfish, degradation, throughput, communication, packets

1. INTRODUCTION

Among various wireless technologies, Mobile Ad hoc Network (MANET) is an active network in the communication. MANET is self-configuring, dynamic and framework-less network. As in Fig. 1 over a wireless link it’s mobile nodes are connected to each other. At any instant of time in MANET, any mobile node has the liberty to meander inside and outside the network. Minimum resource utility, Self-constructive, Distributive and cost consumption are some of the advantages of MANETs. The primary purpose of MANET is to employ itself in a critical application such as military scenarios, emergency purpose, data networks, device networks and commercial sectors etc. High performance is required for MANET to handle the above critical application.

A mobile ad hoc network (MANET) is a continuously self-configuring, infrastructure-less network of mobile devices connected without wires. In a MANET each device is free to move independently in any direction, and will therefore change its links to other devices frequently. Each must forward traffic unrelated to its own use, and it acts as a router. MANETs contains one or more and different transceivers between nodes. MANETs consist of a peer-to-peer, self-forming, self-healing network. Typically they communicate at the Radio Frequency between 30MHZ – 5GHZ.

The practical implementation is limited, Despite the possible use of MANETs in a variety of applications. It’s due to the inherently insecure communication process and non-centralized architecture. Thus providing security in MANETs has become a major concern for researchers. In MANETs, communication between two nodes that are outside each other's transmission range requires a multi-hop process that includes intermediate nodes for data forwarding. These intermediate nodes are independent and most likely candidates to become an attacker node. In MANETs, various attacks caused by malicious intermediate nodes have been reported in literature. These include worm hole, black hole, gray hole, flooding and impersonation attacks, selfish node misbehaving etc. These attacks are performed on UDP based MANETs and their attack methodology tampers the original...
functionality of data communication process making these attacks visible and hence easy to detect. MANETs are more vulnerable to attacks due to: (1) lack of central point for authentication, (2) network management and authorization facility (3) requirement of mutual trust based communication (i.e. multi hop communication) (4) dynamic topology and (5) limited resources.

II. VARIOUS ATTACKS IN MANET

A. **DATA Traffic Attack**

DATA traffic attack deals either in nodes dropping data packets passing through them or in delaying of forwarding of the data packets. As in Fig. 2 Some types of attacks choose victim packets for dropping while some of them drop all of them irrespective of sender nodes. This may highly degrade the quality of service and increases end to end delay. This also causes significant loss of important data. For e.g., a 100Mbps wireless link can behave as 1Mbps.

B. **Black-Hole Attack**

In this attack, a malicious node acts like a Black hole, dropping all data packets passing through it as like matter and energy disappears from our universe in a black hole [2]-[5]. If the attacking node is a connecting node of two connecting components of that network, then it effectively separates the network in to two disconnected components.

C. **Gray-Hole Attack**

A Gray-Hole attack has its own characteristic behavior. It too drops DATA packets, but node’s malicious activity is limited to certain conditions or trigger. Two most common type of behavior: (i) Node dependent attack – drops DATA packets destined towards a certain victim node or coming from certain node, while for other nodes it behaves normally by routing DATA packets to the destination nodes correctly. (ii) Time dependent attack – drops DATA packets based on some predetermined/trigger time while behaving normally during the other instances) Detecting this behaviorist attack is very difficult unless there exists a system wide detection algorithm, which takes care of all the nodes performance in the network. Sometimes nodes can interact with each other and can advise malicious nodes existence to other friendly nodes. This Approach is similar to Black-Hole attack where sequence number feedback might detect some Gray-Hole attack. If multiple paths exist between sender and destination then buffering packets with proper acknowledgement might detect active Gray-Hole attack in progress. But dormant or triggered attack is difficult to detect with this approach.

III. JELLYFISH ATTACK AND ITS TYPES

Jellyfish attack is somewhat different from Black-Hole & Gray-Hole attack. Instead of blindly dropping the data packets, it delays them before finally delivering them [1]. As in Fig. 3 it may even scramble the order of packets in which they are received and sends it in random order. This disrupts the normal flow control mechanism used by nodes for reliable transmission. Jellyfish attack can result in significant end to end delay and thereby degrading QoS [6]-[8]. Few of the methods used by attacker in this attack:

(i) One of the methods is scrambling packet order before finally delivering them instead of received FIFO order. ACK based flow control mechanism will generate duplicate ACK packets which will unnecessarily consume precious network bandwidth and battery life.

(ii) Another method can be, performing selective Black-Hole attack by dropping all packets at every RTO. This will cause timeout in sender node at every RTO for that duration. If nodes use traffic shaping, default flow control mechanism might be triggered to the sender node as it is same as destination overwhelm

(iii) The attacking node can store all the received packets in its buffer but sends them after some random delay maintaining the received packet order. Here also the flow control mechanism gets confused. Sometimes the source node might take a longer route instead of the most obvious shortest route.
A. JellyFish reordering attack

As the name suggests, an attacker node reorders some of the packets before forwarding them. If ACKs of some of reordered packets are not received in time, the sender retransmits them again. From receiver's perspective, each time a packet is received, an ACK is generated. For out-of-order packets, sender shall receive duplicate ACK messages. TCP initiates its flow control mechanism if these duplicate ACK messages exceed a threshold. In our implementation of JF reordering attack, the JF node creates a reordering buffer of size $k$ in its input queue. The data packets in this buffer are reordered before being forwarded.

B. JellyFish periodic dropping attack

In this attack, JF nodes randomly discard some packets over a specified period during communication process. In this way, incorrect route congestion information is conveyed to TCP, which uses dropping of packets as an indication of congestion on the route. The JF-node may either choose to discard a fraction of packets (e.g., 10 packets from every 100 packets) or may discard all the packets received during a slice of time (e.g., discarding data packets for few milliseconds every second near the TCP sender timeout) [9]-[11]. This forces TCP to enter there transmission timeout (RTO) and to increase its RTO value. As the flow becomes stable, attacker repeats the above strategy to sustain the attack and keep the data flow rate low. As JF-node starts discarding packets for some duration, the sender will eventually enter in timeout. TCP handles the timeouts by entering in slow start phase leading to decrease in the network throughput. The throughput decreases as the frequency of packets dropped by the attacker node increases. To maximize the impact of the attack, a JF-node will drop packets as soon as the TCP sender exits its slow start phase. Due to this, the flow will always be in a fragile slow-start state.

C. JellyFish delay variance attack

Round trip time (RTT) of data packets vary considerably due to congestion. Though TCP has a flow control mechanism to adapt to the changes, it cannot determine if the change in RTT is due to dynamic wireless topology, network congestion or JellyFish attack. Also, changes in RTT force TCP to increase RTO. By delaying packets randomly, a JF node can initiate this attack resulting in. Self-clocking of TCP leading to increased collisions and data packet loss. Wrong estimation of the available bandwidth for delay based congestion control protocols such as TCP Westwood and TCP Vegas, Very high RTO estimate, thus decreases network throughput due to delayed detection of congestion in the network. In delay variance attack, JF nodes are selfishly delaying packets. Resultant increase in RTT misleads the sender TCP, which increases its congestion window size and sends traffic in bursts. It will eventually result in more collisions shows an instance of our implemented delay variance attack.

IV. EXISTING SYSTEM AND ALGORITHM

Source node broadcast the data packet to intermediate nodes based on node range and distance algorithm. As in Fig. 4 the source node communicates with the intermediate nodes two-hop away from source node to reach the destination node. The attacker targets the intermediate node and attack the node. The attacker who performs this operation is known as JellyFish attacker. The attack can be of three types namely JF-reorder, JF-delay and JF-drop. This may lead to decrease in throughput.
High performance MANET is required to route data packet between nodes in critical situation. As in Fig. 5 and the table the security algorithm used to improve performance uses various factors. The factors which directly influence the security of MANETs are (i) number of packets generated and forwarded by the neighboring nodes (ii) the past activity of the node (iii) signal strength (iv) delay of forwarding packets and (v) throughput. These factors are optimized to achieve high performance. In existing system, TCP overhead End-to-end delay and degradation in TCP throughput occurs at a certain threshold value. The proposed work improves throughput and the above mentioned parameters by using a new algorithm called Improved Direct Trust-based Detection Algorithm (DMPD). This work will enhance the security of the MANETs.

V. DIRECT MULTIPLE PARAMETER-BASED DETECTION ALGORITHM (DMPD)

---

```
NOTATION | DESCRIPTION
--- | ---
S and D | Source node and Destination node
X | Intermediate nodes
N_i | Number of nodes
N_JF^X | Set of JF nodes identified by X
T_{MIN} | Minimum threshold for trust
A | Attacker JF node
P | Current packet under consideration
T_{nP} | Primary timer at n
T_{nS} | Secondary timer at n
t | Trust values
t_{MIN} | Minimum threshold for t
R_n | Table for trust value
T_i | Time interval
R_i[j] | The trust value for node J as P received by node i
N^JF | Number of JF node
```

---
Though the Direct Trust-Based Detection algorithm isolates and eliminates the JellyFish attacker node from the network, it may produce the false JF-attacker detections since they are monitored directly. As in Fig. 6 In order to overcome this drawback we propose the Direct Multiple Parameter-based Detection (DMPD) Algorithm to increase the throughput, by using an in-direct monitoring method. In which all the neighbours of a node will monitor and share this observation with each other. This approach will decrease the false JF-attacker detections caused by improper overhearing of data packets during promiscuous mode due to the interference and node mobility.

As in Fig. 7 In addition to the parameters considered by DTD, the DMPD considers the following parameters such as, (i) number of packets generated and forwarded by the neighboring nodes (ii) the past activity of the node (iii) signal strength (iv) delay of forwarding packets and (v) throughput.

### A. System Description

Source node multicast request signal (RREQ) to all nodes in the network. Intermediate nodes and destination nodes sends reply signal (RREP) to source node. We proposed a new algorithm called Direct Multiple Parameter-Based Detection (DMPD) algorithm to detect and remove a JellyFish node from an active communication route. In our proposed DMPD algorithm each node uses locally calculated trust values which are collected a time period to identify whether its neighbor node is a JF-attacker or not. To provide more security the DMPD algorithm represents the following parameters: (i) number of packets generated and forwarded by the neighboring nodes (ii) the past activity of the node (iii) signal strength (iv) delay of forwarding packets and (v) throughput. As in Fig. 8 and From the simulation results and the X-graph it proves that the performance of the DMPD algorithm is better than the other existing algorithm. Parameters and it’s formulae are defined as follows,

- **Threshold value = 0.6 (our consideration)**
- **Maximum threshold value = 1**
- **Overall signal strength =**
  \[
  \text{Signal strength of the current node} \times \text{no. of packets dropped}
  \]
- **Past activity of the node =**
  \[
  (\text{past time duration} - \text{current duration}) / 100
  \]
- **Throughput =**
  \[
  \left(\frac{\text{total no. of received packets} \times \text{avg. transmission delay of packets}}{\text{no. of received packets}}\right)
  \]
VI. CONCLUSION

This work introduces a new metric in MANET, To provide more security using the DMPD algorithm which represented the following parameters (i) number of packets generated and forwarded by the neighboring nodes (ii) the past activity of the node (iii) signal strength (iv) delay of forwarding packets and (v) throughput. It increased the efficiency and security compared to the other existing algorithms. And the indirect monitoring method is the greatest boon to this project. The Future enhancement of this work is dealing with the weightage of parameters in the node which is not concentrated in this work.

REFERENCES