Detection of Routing Anomaly using IDS Architecture based on Agents and Clusters in MANETs

D.Srinivasa Rao
Department of Computer Applications
Anil Neerukonda Institute of Technology & Sciences
Visakhapatnam,
Andhra Pradesh, India

T. Pandurang Vital
Department of Computer Applications, Gayatri PG College,
Srikakulam,
Andhra Pradesh, India

T.V.S. Sriram
Department of Computer Applications, Raghu Engineering College
Visakhapatnam,
Andhra Pradesh, India

ABSTRACT
In recent years, the security issues on MANET have become one of the primary concerns. MANETs rely on the cooperation of the nodes participating in the network to forward packets for each other. Therefore, MANET routing protocols are more vulnerable than routing protocols in wired networks. Because of unique features of MANETs, existing security mechanisms, especially Intrusion Detection Systems (IDSs) like authentication and encryption that proposed for wired networks are not suitable for this type of networks. Hence, in this paper, we have proposed a new IDS architecture based on agents and clusters. It detects intermediate nodes misbehaving and anomalies in packet forwarding. Simulation results show that our architecture can achieve low false positive and high detection ratio.

Keywords: Anomaly Detection; Intrusion Detection System (IDS); misbehaving nodes; Mobile Ad hoc Networks (MANETs)

1. INTRODUCTION
Today security is based on the concept of defense-in depth, where multiple layers of defenses are used to prevent network from misbehaving nodes. In the network with high vulnerability such as Mobile Ad hoc Networks (MANETs) which requires strict secure communication, intrusion prevention techniques alone cannot satisfy the security requirements. Therefore, intrusion detection systems (IDSs), serving as the second line of defense, are indispensable for MANETs with high security requirements. MANET IDSs will complement and integrate with existing MANET intrusion prevention methods to provide highly survivable networks [1]. Hence in this paper, we proposed a new architecture based on anomaly and agents. Also, we discuss how to distinct the intrusion and abnormal behaviors from expected and normal behavior. So, we have proposed a quantitative and statistical based algorithm for identifying misbehavior nodes. Our method is a behavioral anomaly based system, which makes it dynamic, scalable, configurable and robust. Finally, we verify our method by running simulations with mobile nodes using Dynamic Source Routing (DSR) protocol as the routing protocol. The rest of this paper is organized as follows. Section 2 presents the network and threat model. Our proposed IDS is presented in section 3. Section 4 presents the simulation and performance evaluation. Finally, in section 5 concludes the paper and discusses some future work.

2. THE NETWORK AND THREAT MODEL

2.1 Network Model
We consider the MANET as a collection of nodes. Each node has an IDS agent for detecting potential abnormalities in packets forwarding process. To reduce the performance overhead of intrusion detection, nodes in a cluster will cooperate to elect a cluster-head node for handling the detection process for the whole cluster. Each cluster-head node is aware of its cluster information. The authenticity of a node is mostly determined by the nodes that are in same cluster. The process of detecting misbehaving nodes can be divided into two steps. First, it needs to determine whether a cluster has a misbehaving node. Should a misbehaving node exist in the cluster then the second step is to locate it [8]. Also, into each cluster, nodes have been organized in a hierarchical structure. In the first level, there is the cluster head, the gateway nodes are in the second level and finally the leaf nodes or ordinary ones are in the third level of this hierarchy.

2.2 Threat Model
Since MANET routing protocols works properly only if the participating nodes cooperate in routing and forwarding, are more susceptible to intermediate nodes anomalies. Hence, in this paper, we will focus on the detection of attacks targeted at MANET routing protocols. So, we use Dynamic Source Routing (DSR) protocol as the exemplary routing protocol to model the behavior of the routing anomalies attacks.

2.2.1 DSR
In DSR when a node has a packet and it does not know the route for the destination, it sends out a route request (RREQ) packet. While this packet is being transferred through the network, all the nodes traversed are recorded in the packet header. A node that knows the route to the destination does not forward the packet further, but appends the route to the route information already accumulated in the packet and returns a route reply (RREP) packet to the source node. After this, the source node maintains the discovered route in its route cache and delivers the packets to the destination node through the discovered route. If any link on a source route is broken, the source node is notified using a route error (RERR) packet [9][10].

3. PROPOSED NEW ARCHITECTURE

In the system aspect, we attach an intrusion detection agent to each node. Its diagram is shown in figure 1.

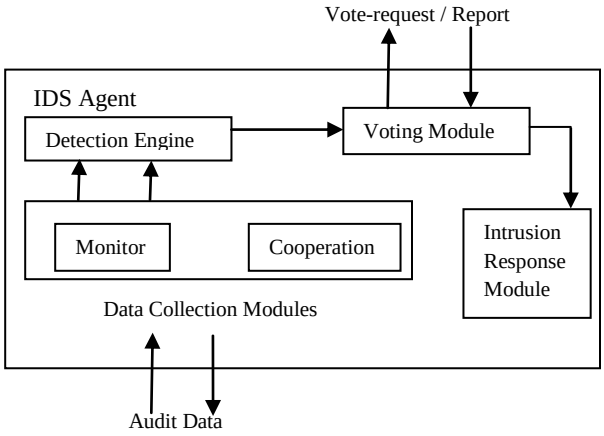

Figure 1. Structure of an IDS agent

3.1 Data Collection Modules

The functionality of the data collection modules is to collect security related data via monitoring local activities and behaviors of neighbor nodes. We define misbehaving nodes as those nodes that have aberrations in data exchange patterns. Also, as mentioned above we define a bucket as a specific count of packets that are transmitted between two nodes [11]. We will have two kinds of buckets: first the long-term or historical bucket which has the statistic information on the quality of packet forwarding of the last N packets; and second the short-term or the data bucket that includes the statistic data on the recently sent M packets. Our suggested architecture uses two modules to gather the local information, which are:

3.1.1 The Monitor Module

We introduce the monitor for monitoring the behavior of its neighbors. Then, monitor based on the behavior of each its neighboring node, a rating is calculated as follow for each of them.

$$RT = \sum \text{packet}_{\text{forwarded}} / \sum \text{packet}_{\text{actual\_received}} - \sum \text{packet}_{\text{destination}} \quad (1)$$

In (1), $\sum$ packet $_{\text{actual\_received,}}$ is the total number of packets that node has received. $\sum$ packet $_{\text{destination}}$, stands for the total number of packets which is destined for the node itself and $\sum$ packet $_{\text{forwarded}}$, is the total number of packets that the node has forwarded to other nodes as an intermediate node.

3.1.2 The Cooperation Module

We present a mechanism that in addition to identifying the misbehaving nodes creates necessary motivation and enforcement for cooperation among nodes and better performance of their tasks in the network. In this mechanism we have considered a privilege of cooperation (CP) for every node which is acquired through the following equation.

$$CP = \lambda \sum \text{packet}_{\text{forwarded}} + \gamma \sum \text{packet}_{\text{destination}} - \varphi \sum \text{packet}_{\text{originated}} \quad (2)$$

In (2), $\sum$ packet $_{\text{forwarded,}}$ stands for the total packets that the node has forwarded for others as the intermediate node.
$\sum$ packet $_{\text{destination}}$ is the total packets that node as a destination node has received and $\sum$ packet $_{\text{originated}}$, stands for the whole packets that are generated by the node as the source node and injected to the network. Since the expenses (according to the rate of energy, CPU and memory consumption) of every of this operation are not the same, we have considered this fact in our CP computation. In this formula, λ stands for the expense and weight of forwarding every packet to other nodes as the intermediate node, γ stands for the expense and weight of every received packet by the node as the destination node, and φ stands for the expense and the weight of every packet that the node has generated as the source node. Each node keeps two tables called RT and CP that each entry of these tables contains RT and CP values of neighboring and some other nodes that belong to the same cluster and are along the communication path segment, respectively. At the end of every short-term bucket, data collection modules send these tables to detection engine.

3.2 Detection Engine

Detection engine calculates a table called SNB (Score of Nodal Behavior). Each node keeps this table that each entry of this table contains SNB value of neighboring and some other nodes that belong to the same cluster and are along the communication path segment. Detection engine calculates the SNB function of each node based on two functions RT and CP as follow.

$$SNB_i = \mu * RT_i + \delta * CP_i \quad (3)$$

Where $0 < \mu, \delta < 1$.
In (3), the coefficients of μ and δ are the weight of functions RT and CP, respectively. Since the computation of the SNB function has been repeated periodically during the lifetime of a node, it gives us good information on behavior and operation of the node. In each SNB table entry besides the values of SNB, contains the timestamp information to indicate the time when the SNB value was last updated. The task work of the detection engine is to detect the misbehaving nodes. The detection of misbehaving nodes requires an exact definition of the term misbehaving. It also requires specifying an appropriate threshold between normal and abnormal behaviors. In this article, we have defined the misbehaving nodes as the nodes which had abnormality in the process of packets forwarding. Since, we have used cluster structure with a cluster-head for each of them, each cluster-head determines a threshold for its cluster as follow, and it updates it based on the amounts of SNB of all its nodes. Then at the end of each short-term bucket, it sends this threshold to all its cluster nodes.

$$Th = \tau \times \frac{1}{|N_{SNB}|} \sum i \in N_{SNB} S_i \quad (4)$$

Where $0 < \tau < 1$.
In (4), N_{SNB} shows all listed nodes in the SNB table of the cluster-head node, $|N_{SNB}|$ is the number of the nodes, and Si shows the value of SNB related to the node *i*. If the detection engine finds one or some values of SNB in the table that are

less than the threshold, then it realizes that there may be one or some misbehaving nodes in its cluster. So it sends to the voting module a vote request about the suspect node(s).

3.3 Voting Module

The operation of voting module depends on its node type. If a node is a leaf or gateway one, it only sends the alarms (vote-request) and reports generated by detection engine to the cluster-head node. While if the node is a cluster-head node, then the voting module handles the received alarms and reports from leaf and gateway nodes. Also, voting module of the cluster-head node allows the voting or prevents it by aggregation and correlation of the received alarms and reports along with its own information, and also by considering the privilege of the suspect and the node requesting for voting. If the cluster-head node agrees with voting, then it announces the process of voting by broadcasting a packet called vote-request packet to all its cluster nodes and sends the result of voting process to its cluster nodes. Finally, voting module of cluster-head sends the calculated threshold to its cluster nodes.

When the voting module of each leaf or gateway node receives the vote request packet from its cluster-head node, votes for or vetoes the suspect node according to the results announced by the detection engine and send result to the voting module of its cluster-head node. In the process of voting, simply accounting the positive or negative votes is not fair, because the values of each node's SNB are not the same throughout a cluster. The values with recent timestamp are more important than the values which are older. We have considered this fact by considering the *w* variant as the time weight. Let's assume that k nodes participate in the process of voting for or against the authenticity of the node *m*. The voting module of the cluster-head node calculates the result of voting according to below.

$$V_m = \frac{k}{N} \times \sum_{i=1}^{k} w_{im}\, s_{im}\, v_{im} \qquad (5)$$

In (5), S_{im} shows the SNB value of the node *m* in the SNB table of node *i* and N shows the total number of nodes
Which are located within the cluster. w_{im} shows the time weight of S_{im}. v_{im}=1, if node *i* votes for node *m*, and v_{im}=-1, if node *i* votes against node *m*. At the end of voting process, voting module of the cluster head node sends the result of voting (V_m) to its intrusion response module and voting modules of all its cluster nodes.

3.4 Intrusion Response Module

According to the results of voting, the *m* node is a well behaving one and is acquitted or it is a misbehaving one and should be punished. If $V_m << 0$, then intrusion response module finds that the node *m* is misbehaving and adds *m* to its blacklist. When a node is added to the blacklist, the intrusion response module deletes it from its SNB and routing tables. Thereafter, the intrusion response module prevents the node from cooperating with the blacklisted node. If $V_m >> 0$, then the intrusion response module realizes that the node *m* is a well behaving one, and those nodes that have voted against it, should update the amount of their SNB. Let's assume that there are *K* nodes in the communication path segment, then the nodes that have voted against *m*, update the amount of their SNB as follow.

$$s_m = \frac{1}{k} = \sum_{i=1}^{k} s_i \qquad (6)$$

In (6), S_i stands for the amount of the SNB related to the node *i* in the communication path segment inside the cluster. This equation decreases the difference among the voting nodes and prevents from repeated requests for voting in near future. If the value of $V_m \cong 0$, then the process of voting is repeated in the clusters level. In this situation, the cluster head node carries out the process of the voting in a vast area by sending vote request to the cluster-head nodes of the downstream and upstream (if there is any).

4. SIMULATION AND PERFORMANCE EVALUATION

GloMoSim 2.03 [12] simulator is used to simulate our model. We conducted our experiment using Dynamic Source Routing (DSR) protocol as the routing protocol. The channel capacity of mobile hosts is set to 2 Mbps and the transmission range is set to 250 meters. A free space propagation model with a threshold cutoff is used as the channel model. We use the Distributed Coordination Function (DCF) of IEEE 802.11 for wireless LANs as the MAC layer protocol. In the simulation, mobile nodes move in a 2000m X 2000m region. The mobility model is the random waypoint model. The minimal speed is 5 m/s, and the maximal speed is 15 m/s. various source-destination pairs are selected randomly to generate Constant Bit Rate (CBR) traffic as the background traffic. The size of all data packets is set to 512 bytes. The duration of each simulation was 1800 seconds.

4.1 Simulated Attacks

In this article we have simulated three flooding, black- hole and gray-hole attacks to evaluate the functionality of our IDS. The functionality of these IDS is shown in different graphical format.

4.1.1 Flooding attack [13]

In this attack, the misbehaving node pumps a great deal of useless and garbage packets to the network. In this way it corrodes the resources of the network (like bandwidth and energy).

4.1.2 Black-hole attack [14]

In this attack, a misbehaving node uses the routing protocol to advertise itself as having the shortest path to the node whose packets it wants to intercept. The attacker will then receive the traffic destined for other nodes, then can drop or modify the packets.

4.1.3 Gray-hole (selective forwarding) attack [15]

This attack is a special kind of black-hole attack. But contrary to the black-hole attack, the gray-hole attack, attacks the packets based on a probability function or it does it by attacking specific nodes' packets, and not all the received packets.

4.2 Performance Metrics

Detection Ratio: It is defined as the percentage of IDS capability in detecting the misbehaving nodes; and is resulted from dividing the accurate detections into all detections. False Positive Ratio: It is defined as the percentage of decisions in

which well behaving nodes are flagged as misbehaving ones inaccurately.

4.3 Simulation Results

In the first simulation, we have considered the relation between the detection rate and number of the nodes. As it is showed in figure 2, most of the attacks have been detected successfully. In detecting three attacks, black-hole, gray-hole and the flooding attack, our IDS have had a good stability, and its detection rate has been over 84%. But, detection rate of these attacks has decreased in high density networks. This was predictable, because in high densities, the amount of the traffic is high and it is not possible to distinguish between the network's normal status and it's under attack situation easily. For example, a node cannot understand the package being deleted because of the attack or due to congestion.

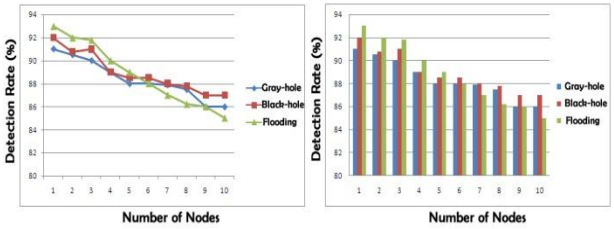

Figure 2. Detection rate vs. number of nodes

The result of the simulation between false positive rate and the number of nodes is shown in figure 3. In its worst status, the false positive rate has not been over 14%, this is a good rate. In networks with high densities, the false positive rate is high and this is so natural. For example, in networks with high densities, the amount of the generated traffic is high. So it is not possible to identify whether the high rate of traffic is due to flooding attack or then normal situation of the network. In this condition, the sending nodes whose packets have been dropped because of reasons other than attack have to resend them. So the amount of SNB function of these nodes decreases and they are considered as misbehaving nodes inaccurately.

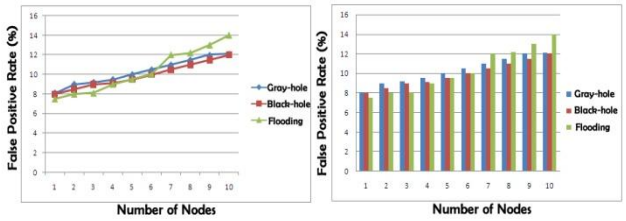

Figure 3. False positive rate vs. number of nodes

In the reminder simulations we have investigated effect of percentage of misbehaving nodes on the performance of our IDS. The number of nodes in these simulations is 60. In figure 4, we have shown the result of the simulation between the detection rate and the percentage of misbehaving nodes. As it is seen, most of the misbehaving nodes have been detected successfully, but as it was predicted, in networks with high percentages of misbehaving nodes, the detection rate decreases. This happens due to different reasons and the most important of all, is related to the process of voting. Because, when the percentage of misbehaving nodes is high the accuracy of voting result will be reduced.

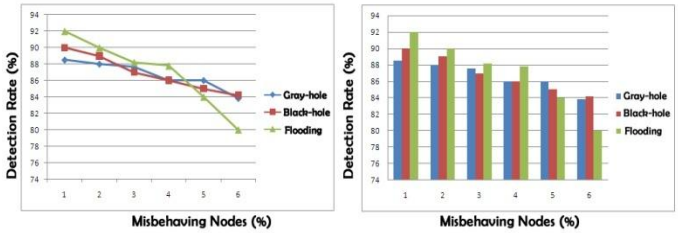

Figure 4. Detection rate vs. percentage of misbehaving nodes

The result of the simulation of the relation between false positive rate and the percentage of the misbehaving nodes is shown in figure 5. As the result of the simulation shows, when at first the percentage of the misbehaving nodes is low, the false positive rate is high; this was predictable because in normal situation, some packets may not reach the destination due to reasons like congestion or link destruction. This situation is mistaken for misbehaving of the node. But when the percentage of the misbehaving nodes is high, the false positive rate again increases, because as we mentioned in the previous paragraph, under this situation the exactness of the voting process decreases.

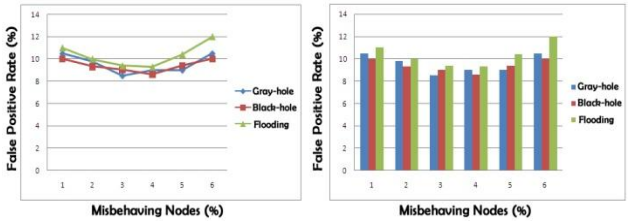

Figure 5. False positive rate vs. percentage of misbehaving nodes

5. CONCLUSION AND FUTURE WORK

We have presented a quantitative approach to detect misbehaving nodes in MANET. The approach is based on agent and clusters. Using the routing attacks as the threat model and DSR protocol as routing protocol, we have carried out simulation and simulations have been conducted to verify the effectiveness of the approach. One of our future works is to develop our architecture for other layers. We also simulate more attack scenarios to investigate efficiency of our approach.

6. REFERENCES

[1] B. Sun, K. Wu, and U. Pooch, "Routing Anomaly Detection in Mobile Ad Hoc Networks," Proc. of the 12thInternational Conference on Computer Communications and Networks (ICCCN'03), Dallas, TX, Oct. 20-22, 2003.

[2] . Mishra, K. Nadkarni, and A. Patcha" Intrusion Detection in Wirelessad Hoc Networks" IEEE Wireless Communications, 2004, pp. 48–60.

[3] Y. Zhang and W. Lee, "Intrusion Detection in Wireless Ad-hoc Networks," Proc.of the MOBICOM 2000, 2000, pp. 275–283.

[4] Y. Zhang, W. Lee, and Y. Huang, "Intrusion Detection Techniques for Mobile Wireless Networks," Wireless Networks, vol. 9, 2003, pp 45-556.

[5] O. Kachirski, and R. Guha, "Intrusion Detection Using Mobile Agents in Wireless Ad hoc Networks," Proc. of the IEEE Workshop on Knowledge Media Networking, July 2002, pp.153-158.

[6] P. Albers, O. Camp, et al. "Security in Ad Hoc Networks: a General Intrusion Detection Architecture Enhancing Trust Based Approaches," Proc. of the 1st InternationalWorkshop on Wireless Information Systems (WIS-2002), April 2002, pp. 1-12.

[7] B. Sun, K.Wu, and U. W. Pooch, "Alert Aggregation in Mobile Ad Hoc Networks," Proc. of the ACM Workshop on Wireless Security (WiSe'03) in conjuction with the 9th Annual International Conference on Mobile Computing and Networking (MobiCom'03), 2003, pp. 69-78.

[8] T. Li, M. Song, and M. Alam, "Compromized Sensor Node Detection: A Quantitative Approach," Proc. of the IEEE International Conference on Distributed Computing Systems, 2008, pp. 352–357.

[9] S. R. Das, C. E. Perkins, and E. M. Royer," Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks," Mobile Ad Hoc Networking Working Group, Internet Draft, February 2001.

[10] E. Çayırcı and C. Rong, "Routing in Ad Hoc, Sensor and Mesh Networks", in Book Security in Wireless Ad Hoc and Sensor Networks (chapter 5), CRC Press LLC, 2009.

[11] K. Kumar, "Intrusion Detection in Mobile Adhoc Networks," Master's Thesis, University of Toledo, December 2009.

[12] GloMoSim: Global Mobile Information Systems Simulation Library, http://pcl.cs.ucla.edu/projects/glomosim/.

[13]C. Hongsong, F. Zhongchuan, W. Chengyao, J. Zhenzhou, and H. Mingzeng, "Using Network Processor to Establish Security Agent for AODV Routing Protocol," Journal of Computing and nformation Technology - CIT 15, 2007, pp. 61–70.

[14] F. Anjum and P. Mouchtaris, Security for Wireless Ad Hoc Networks (Chapter 1), CRC Press LLC, 2007.

[15] E. Çayırcı and C. Rong, "Security Attacks in Ad Hoc, Sensor and Mesh Networks," Security in Wireless Ad Hoc and Sensor Networks (Chapter8), CRC Press LLC, 2009.