TIDS: Trust-Based Intrusion Detection System for Wireless Ad-hoc Networks

Novarun Deb and Nabendu Chaki

University of Calcutta
novarun.db@gmail.com, nabendu@ieee.org

Abstract. This paper aims to propose a new trust-based Intrusion Detection system (IDS) for wireless, ad-hoc networks with or without mobility of nodes. In fact, the proposed solution not only detects intrusions but also proactively responds towards route setup avoiding the compromised nodes. It could be extended for mesh or hybrid networking environment too. Trust is evaluated as the weighted sum of direct evaluation of the neighboring nodes as well as from the indirect references. A sliding window is defined on the time scale and the IDS is to be evoked after every time slice. Indirect reference is derived from the recommendations of those 1–hop neighbors of the target node that are also neighbors of the evaluating node. The performance of the proposed algorithm has been evaluated using the Qualnet network simulator. Simulation results also establish superiority of the proposed algorithm over HIDS, another recent trust-based IDS for wireless ad-hoc network.

Keywords: Greyhole attack, Denial of Service attack, RREQ packet, RREP packet, Trust Request packet.

1 Introduction

Wireless and cellular networks have tremendously grown over the last four to five years. New technologies have been deployed in this domain with MANETs and Wireless Mesh Networks among the most notable ones. Sensor networks are also finding major applications such as Border Area Surveillance or Disaster Recovery Management. Also, end–user requirements have resulted in cellular and mobile networks being exploited to their fullest. Millions of applications are being used by customers that inherently demand security. This is where Intrusion Detection System plays a very important role. Most wireless network technologies have energy constrained nodes. Consequently, computation intensive procedures are often avoided. Thus, unlike traditional hardwired networks, intrusion prevention is not at all an option for wireless ad–hoc networks as these are quite computation intensive.

Intrusion detection is one of those safety mechanisms that is energy-efficient as well. Also, more effective is an Intrusion Response System that takes some corrective measures once an intruder is detected. This paper aims to propose a new Intrusion Detection and Response System for Wireless ad-hoc networks, in general. The proposed solution not only detects intrusions during application traffic but can also be

A. Cortesi et al. (Eds.): CISIM 2012, LNCS 7564, pp. 80–91, 2012.

proactively involved towards route setup. The most trusted route will be set up for better QoS. Trust is the basis of the proposed IDS. Several recently proposed trust models have been studied and reviewed in the state of the art section. All these models have certain shortcomings and are vulnerable to attacks under certain situations. Also, none of these trust models have been extensively tested on any network simulator for any type of comprehensive results. As part of the paper, a trust model has been proposed from which an intrusion detection algorithm has been designed.

The rest of the paper is structured as follows. Section 2 describes the State of the Art Review on Trust based IDS. Section 3 discusses in details the working of the Trust based algorithm (TIDS). Section 4 highlights the simulation results of our algorithm and compares its performance with another IDS algorithm HIDS[6]. Section 5 concludes the paper with Section 6 Acknowledgements and Section 7 listing the references.

2 State of the Art

Various models have been proposed for sharing resources in a P2P environment. Quite often, these models fail to consider the trust of peers prior to resource sharing. PET [1] is a one of the highly sited trust models where a peer always trusts itself. Trust on a peer increases slowly but decreases rapidly. In [1], trust is evaluated quantitatively as the combination of two components – reputation and risk. Reputation is a long term assessment of the behavior of the peer in the past. Risk on the other hand is a short term assessment of the peer's most recent behavior. Reputation component of trust comprises of two components – recommendation and direct interaction. Recommendations dominate trust evaluation when there has been no direct interaction in the past. A weighted evaluation of these components is used in evaluating Reputation. Direct interaction information is also used for evaluating the Risk component of Trust. PET classifies peers based on the QoS provided by them. Four major categories of QoS are Good, No Response, Low Grade, and Byzantine behavior. Nodes are rewarded positively for Good behavior only. Nodes are negatively rewarded for the other three categories. The magnitude of negativity decreases from Low Grade through No Response and Byzantine behavior. Risk is evaluated as the amount of negative score earned due to bad services by the peer in a specific time interval.

In [2], Cho et. al. have proposed trust management for MANETs using trust chain optimization. Trust is evaluated based on four components – residue energy level and co-operation (QoS Trust) and honesty and closeness (Social Trust). The trust value of a node *i* is evaluated by a node *j* as the weighted sum of these four components. Residue energy level and honesty trust component values are binary, co-operation trust component is a probabilistic value based on the node's behavior in the last update interval, and closeness component is an integer representing the number of 1-hop neighbors of a node. Every node evaluates trust of its 1-hop neighbors by observing its behavior to packet forwarding. Trust evaluation is broadcast throughout the network in the form of status exchange messages.

Li Xiong and Ling Liu proposed a new trust model in PeerTrust [3]. PeerTrust computes the trust of peers in a network as a function of 3 components. First, a node N becomes trustworthy when other peers who have interacted with N find it to behave normally. Second is the context of satisfaction. It defines the total number of interactions that a node has performed with its peers. Finally the Balance factor of trust is used to reduce the effects of incorrect satisfaction information coming from malicious nodes. A trust metric *T(u)* for node u is computed as the total satisfaction earned by *u* and multiplied by the balance factor of each peer and averaged over the total number of interactions that u has participated in. However, PeerTrust fails to capture the most recent malicious behavior of highly reputed nodes. This is taken care of by specifying a sliding window on the time scale. PeerTrust uses the P-Grid algorithm for distribution and aggregation of trust data across a P2P network. A key value is assigned to each peer based on its ID. Each node stores and maintains trust data about one or more peers in the network. As peers can behave maliciously, any intentional false trust data about a peer gets replicated in the local databases of more than one peer. This redundancy has its overhead. Such malicious behavior could be avoided by following a voting by consensus algorithm.

Wang, Mokhtar and Macaulay proposed a trust model based on the concept of H-index. C-index [4] incorporates the past experience a peer node has had with a collaborator. The more the number of trustworthy recommendations from a peer node, the higher should be the credibility of its recommendations. Also, trust models should consider the diversity of trustworthy collaborations. The larger the number of peer nodes with which a node collaborates, the greater is the reliability of its recommendation. Trust Depth in a community of nodes is measured as the number of Pure Positive Feedbacks (PPF) a node receives from its peer. It is defined as the difference between the number of satisfactory and unsatisfactory feedbacks from that node. Trust Breadth is the number of peers from which a node receives at least one PPF. Based on TD and TB, the C-index of a node is evaluated. The C-index of a node is used in evaluating its trust. It is defined as the number of peers (Z) in a community of N nodes which have sent at least 'Z' PPFs to the node. The C-index mechanism of trust measurement is much more robust as it is immune to attacks as any single node sending multiple PPFs to a node does not affect its C-index. However, the method remains vulnerable to synergistic attacks. The C-index mechanism fails when the number of attackers is larger than the current C-index of a node.

In [5], Luo, Liu, and Fan have proposed a trust model based on fuzzy recommendation for MANETs. Trust is defined by 3 components – past experience, current knowledge about the entity's behavior, and recommendations from trusted entities. The Fuzzy Trust Model centers around a parameter called the Local Satisfaction Degree (S_{ij}). S_{ij} is the difference between the number of successful and unsuccessful transactions between two nodes *i* and *j*. The Fuzzy Indirect Trust Model is the generic trust model that evaluates trust from two component values – Direct Trust and Recommendation Trust. Direct Trust is evaluated by a node on its neighbor as a result of the interactions between them. Recommendation Trust depends on the recommendations provided by a neighbor about a distant node. Recommendation Trust is evaluated by a node transitively or by consensus. It is evaluated as the combination of

the Recommendation from the neighbor and the Direct Trust that the node has on that neighbor. The neighboring node makes a recommendation about the distant node based on what it receives from its neighbors, transitively. The node has Direct Trust evaluated for all its neighbors and each neighbor makes a Recommendation about the distant node. Consensus Recommendation Trust is the union of all these trust recommendations. However, recommendations from a highly trusted node remain questionable (e.g. synergistic effect of selfish nodes). Thus, trust value of a node is computed globally by combining recommendations from all nodes. RFS-Trust uses an adjusted cosine similar function to find the similarity between nodes i and j. The higher the degree of similarity, more consistent is the evaluation of trust between the respective nodes as compared to other nodes in the network. Thus, it is not a high range of trust values that makes a node's recommendation credible. Rather, credibility of recommendations increases with similarity in rating opinions.

3 The Proposed Trust-Based IDS (TIDS)

Intrusions need to be detected under varying circumstances. This paper focuses on two such scenarios where intrusion detection becomes essential. Since the proposed algorithm is Trust based, intruders are identified on the basis of their trust values. Intrusion detection is essential during route setup. Good Quality of Service can be ensured only when the most trusted route is setup between the source and the destination. Thus, trust value of nodes has to be considered when Route Request and Route Reply packets are being exchanged. The dynamically changing topology of the mobile ad-hoc network causes the routes between them to change frequently. In such a scenario, intrusion detection is even more important as nodes may change their behavior over time. As long as packets are being sent along a particular route, some intermediate nodes may start behaving selfishly or maliciously. In order to detect such intruders, the IDS algorithms are to be evoked at regular intervals. The network should react differently for destination nodes and intermediate nodes. Whenever a destination node is found to be an intruder, the application is terminated and the destination node is blacklisted. If an intermediate node is found to be an intruder, it is bypassed and the route is re-established. The malicious node is also blacklisted.

Before getting into the details of the IDS, let us consider some of the common attacks. The most commonly simulated attack in networking journals is the blackhole attack where a node drops all the packets that are sent through it. However, considering the fact that attackers are intelligent enough, a more practical and realistic attack is the greyhole attack or selective forwarding. Here, a node behaves as a good node to increase its reputation within the network. Once it becomes highly reputed, it starts dropping packets. Later, it again increases its reputation and prevents itself from being detected. There is also the Denial of Service (DoS) attack that can be implemented in more ways than one. The motive behind DoS attack is to consume the resources of the network so that peers are denied service. Detecting spurious packet generation would become all the more difficult if the DoS agent is a member on the route from the source to destination of some application. A stand-alone node that generates

spurious packets can be easily detected. Thus, it is assumed that both greyhole attackers and DoS agents will be on the route from the source to the destination of some application. Rather, during route setup, these attackers will ensure that a route is set up through them by returning corrupt reachability information about the destination. Keeping these attack scenarios in mind, one can conclude that intrusion detection needs to be done only for the nodes that lie on the route between a source and a destination of some application. Nodes that are not part of active applications will not be a part of the intrusion detection as well. This makes the proposed intrusion detection algorithm a lightweight process. All nodes in the network need not execute the intrusion detection algorithm redundantly.

3.1 Working Principle

Intrusion detection is mandatory during the route setup process. The solution proposed in this paper is based on the following working principles.

- Every node maintains trust information about its 1 – hop neighbors.
- Trust is evaluated as the weighted sum of 2 components – *Direct Valuation* and *Indirect Reference*.
- Direct Valuation is again a function of 2 factors – *Reputation* and *Risk*. Reputation is the measure of the long term evaluation of the behavior of a node. Risk is the valuation of the most recent behavior of the node.
- A sliding window is defined on the time scale. The Intrusion detection algorithm is executed after every time slice.
- Indirect Reference refers to the recommendations from 1 – hop neighbors of the "target node" which are also neighbors of the "valuation node".

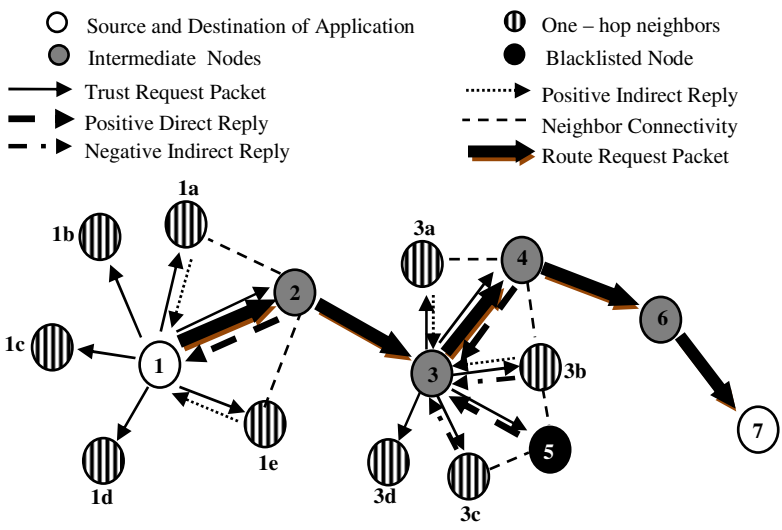

Fig. 1. The Route Request Mechanism

Fig.1 illustrates how the proposed intrusion detection algorithm works during the route setup process. Route Request packets are initiated from the source of an application. The *source node* '*1*' sends a Trust Request Packet to all its one-hop neighbors – *1a, 1b, 1c, 1d, 1e,* and *2*. Every node replies with Direct Valuation of itself and Indirect References about one-hop neighbors which are common to itself and the source of the Trust Request packet. Here the source node '*1*' receives replies from *1a, 1e,* and *2*. It is obvious that intruders will speak highly of themselves. Also, attackers can provide incorrect trust information about nodes in their efforts to establish routes through themselves. Thus, the source of the Trust Request packets does not believe the responses coming from its one-hop neighbors blindly. Since every node maintains trust information about its one-hop neighbors, the source associates a credibility factor with the replies coming from its neighbors. After trust evaluation, *node 2* is found to be the best node between the source and the destination.

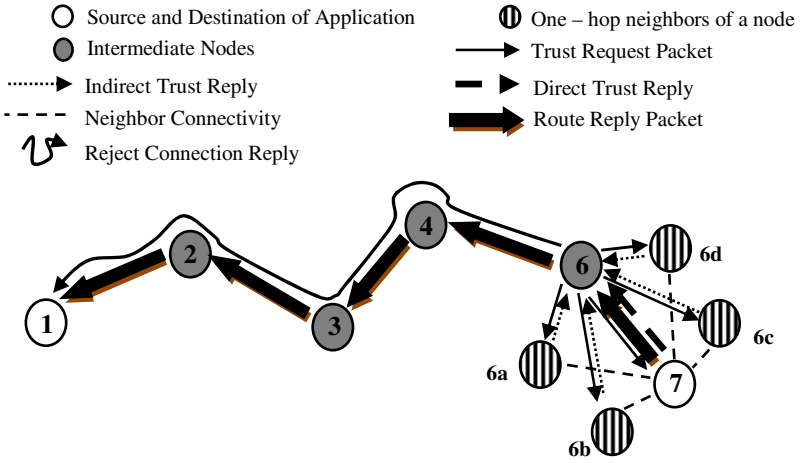

Fig. 2. The Route Reply Mechanism

This procedure is repeated at every node. The figure illustrates another scenario. When the Route Request packet comes to *node 3*, the same procedure is repeated as above. The one-hop neighbors of *node 3* - *3a, 3b, 3c, 3d, 4,* and *5* – reply to the Trust Request coming from *node 3*. *Node 3* gets positive replies about *node 4* but negative replies about *node 5*. *Node 3* associates the credibility of these replies coming from its one-hop neighbors. It evaluates *node 4* to be the most trustworthy and *node 5* as an intruder. Thus, the Route Request is forwarded in the direction of *node 4*. This procedure gets repeated until the Route Request reaches the destination node.

Fig.2 illustrates the procedure when the Route Request reaches the destination. When the Route Request reaches *node 6*, it sends Trust Request packets to all its neighbors – *6a, 6b, 6c, 6d,* and *7* – including the destination. The destination replies with a Route Reply and also mentions the number of its one-hop neighbors in the Route Reply message. All those one-hop neighbors of *node 6* that are also neighbors

of the destination return their trust information about the destination. *Node 6* evaluates the trust of the destination and decides whether to forward the Route Reply to the Source or to return a Connection Abort message.

3.2 Intrusion Detection and Rerouting

Intrusion detection also becomes essential as a part of maintenance. Once connection has been established between the source and the destination, application traffic starts flowing between the two. An intruder may start behaving maliciously or selfishly at some random time instant. Trust evaluation begins at the source. The source evaluates the trust of its one – hop neighbor which is on the route to the destination. Once trust is evaluated for the one – hop neighbor on the source – destination route, the same procedure is repeated for the next node on the route. This continues till the trust value of the destination is evaluated.

If during this process, a node detects its peer on the source – destination route to be behaving *selfishly* or *maliciously*, then the rerouting mechanism is initiated. Figure 3 best illustrates this mechanism. Suppose the existing route for application traffic is through 1→2→3→4→6→7. During intrusion detection, *node 3* finds that *node 4* has been behaving in a malignant manner. *Node 3* discards the existing route and tries to reroute traffic to the destination bypassing *node 4*. It finds *node 5* as trusted and reestablishes connectivity with *node 6* via *node 5*. Thus, the newly established route for application traffic becomes 1→2→3→5→6→7.

If intrusion detection for maintenance finds that the destination has become malicious then the application is closed.

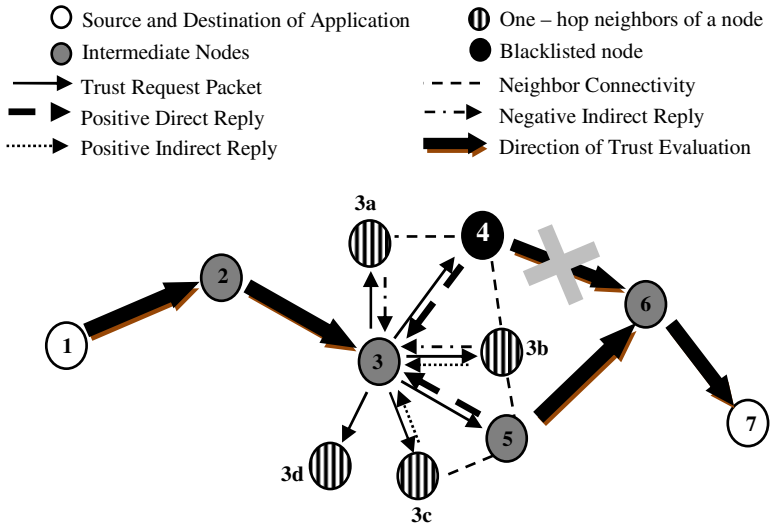

Fig. 3. The Rerouting Mechanism

3.3 The Trust Model

The entire process of routing and intrusion detection is based on the trust evaluated by a node for its one – hop neighbors. The trust model has two main underlying concepts – *Direct Valuation* and *Indirect Reference*.

Direct Valuation is a measure of how the node evaluates the Trust of its one-hop neighbors. Every node monitors the packet forwarding behavior of its one-hop neighbors. A benign node should forward all the packets that it receives from its previous hop neighbor. Thus, packet arrival rate (PAR) and packet delivery rate (PDR) play a decisive role in deciding the behavior of a node. For normal node behavior PAR and PDR tend to be equal. In other words, PAR – PDR tend to zero. Keeping in mind wireless network constraints like mobility and link failure, the normal behavior of a node is classified when the difference (PAR – PDR) lies within a given threshold.

In the selective forwarding attack scenario, a node drops packet occasionally. At other times, it behaves like a normal node. Normal behavior of a node is positively rewarded by increasing that node's trust value. Thus, occasional malicious behavior becomes even more difficult to detect. The proposed IDRS addresses this issue using two separate measures for *Risk* and *Reputation*. Risk is a measure of the node's behavior in the last time slice since the last time the intrusion detection algorithm was run. Reputation is the measure of the long term behavior of a node. Classifying Direct Valuation into Risk and Reputation helps in identifying the most recent behavior of a node in contrast to its long term behavior on the time scale.

Since Direct Valuation depends on the PAR and PDR information coming from one-hop neighbors, attackers may easily tamper this information. Thus, trust of a node is not updated solely on the basis of Direct Valuation. One also needs to consider the reputation of the target node to all its one-hop neighbors. Thus, Indirect References are considered from all those one-hop neighbors that are common to both the evaluation node and the target node. Thus, Indirect Reference of the evaluation node consists of the Reputation information coming from all those one-hop neighbors which are also neighbors of the target node.

Every node maintains a Packet Receive (PR) and Packet Send (PS) counter. After every time slice, these counter values are sent to the node's one-hop neighbors. The neighbors keep a track of the Reputation of the node by summing the PR – PS values coming at the end of each time slice. Also the value of the PR – PS counters in the last time slice measures the Risk. When a node receives a IDS Request packet from an evaluation node, it sends its PR – PS counter values, and Reputation and Trust information. The PR – PS values of the target node is used to evaluate the Risk. These values are summed up with the existing Reputation data and Reputation information of other one – hop neighbors become the evaluation node's Indirect Reference information. These three measures are combined to evaluate the reward for the target node's behavior in the last time slice as follows:

$$\text{Reward} = (W_1 \times \text{Risk}) + (W_2 \times \text{Reputation}) + (W_3 \times \text{Indirect Reference}) \quad (1)$$

The above formula is used to generate negative rewards by assigning negative weights to W_1, W_2, and W_3. Also, these weights are normalized so that $W_1 + W_2 + W_3$

= -1. This formula will be used only when the target node has behaved maliciously in the last time slice, i.e., abs(PAR – PDR) > Threshold. For normal behavior, the Reward generated is positive as follows:

$$\text{Reward} = (\text{PAR} + \text{PDR}) / 2 * W_4 \quad (2)$$

W_4 is chosen so that Positive reward is not very large. Nodes must not be able to increase their trust values rapidly by behaving normally in some time slices. Once the reward for a node is appropriately calculated, the trust value of the node is updated as follows:

$$\text{Trust}(t) = \text{Trust}(t\text{-}1) + \text{Reward} \quad (3)$$

Based on the above formula, the trust of a node may increase gradually or decrease rapidly. Once the trust value of a node is updated, it is checked whether the trust value falls below a certain threshold. If so, then the node is classified as an attacker.

3.4 Algorithm for Intrusion Detection during Route Setup

1. The source initiates route discovery by generating RREQ packets.
2. Whenever a node receives a RREQ packet it forwards the packet to the most trusted one-hop neighbor on the route to the destination.
3. The node broadcasts Trust Request packets to its one-hop neighbors.
4. All neighbors reply with packet forwarding information about itself and Trust information about their one-hop neighbors.
5. The source of the Trust Request packet evaluates the trust of all its one-hop neighbors.
6. The most trusted neighbor is forwarded the RREQ packet.
7. If a node finds the destination to be its neighbor, it forwards the RREQ packet to the destination.
8. Trust value of the destination is evaluated by the pre-destination node.
9. The destination responds with an RREP packet. Depending on the trust evaluated of the destination, the pre-destination node either forwards the RREP packet or returns a “Cancel Application” message towards the source.

3.5 Algorithm for Intrusion Detection as Part of Maintenance

1. After time slice expires the source initiates the Intrusion Detection Algorithm.
2. Source sends Trust Request Packet (TRP) to its 1- hop neighbors.
3. The 1 – hop neighbor on the route to the destination returns its Packet Forwarding information. This is the Direct Valuation data.
4. Those 1 – hop neighbors which are not on the route to the destination, check if the “target node” is their 1- hop neighbor.
5. If so, they return trust information about the target node to the source of the TRP. This is the Indirect Reference.
6. The sender of the TRP receives Direct and Indirect Information.

7. Reputation is the trust value of the target node currently available at the source of the TRP. Risk is the Packet Forwarding information returned by the target node for the last time slice.
8. Indirect recommendations coming from other 1 – hop neighbors are accumulated, averaged and combined with results from the previous step.
9. If the target node is found to be an intruder, then a WARNING message is sent to the source of the TRP that the route is no longer safe.
10. Whenever an intermediate node receives such a message it reestablishes a new route from itself to the destination.

4 Simulation Results

The proposed IDS has been successfully implemented using the standard Network Simulator - QualNet. In this simulation, some nodes have been arbitrarily initialized with higher trust values compared to other nodes. The proposed mechanism successfully sets up routes through the highly trusted nodes. Both Greyholes and DoS agents have been implemented as having high initial trust values. This is practical as attackers do try to attain high trust among their peers before launching an attack. The trust value of course changes dynamically during simulation. The data points collected reflect the sensitivity of TIDS compared to HIDS under similar conditions.

4.1 Simulator Parameter Settings

In order to compare the performance of the proposed solution in terms of Intrusion Detection, HIDS [6], another recent trust based IDS, has also been simulated under the same environment settings. The proposed TIDS solution is compared with HIDS to compare different attacks like *Greyhole,* and *Denial of Service (DoS).* Table 1 describes the parameters with which we have simulated the proposed TIDS. Trust value nodes vary from 0 – 16. Trust value of 6 is the threshold value below which a node is detected as an intruder. All normal nodes are initialized with a threshold value of 6. Certain nodes can have higher trust. The .config file has been suitably modified to assign a trust value of 10 to these highly trusted nodes.

Table 1. Simulator parameter settings

Parameter	**Value**
Terrain area	1500X1500 m^2
Simulation time	200 sec
Mac Layer protocol	DCF of IEEE 802.11b standard
Network Layer protocol	AODV routing protocol
Traffic Model	CBR
Number of CBR applications	10% of total nodes
Highly Trusted Nodes	Randomly selected
IDS time slice	10 sec

4.2 Simulation Results

The following data were collected based on the above simulator settings. Four sets of data were collected. The average of this is taken for comparative analysis of performance against HIDS. Data is taken with respect to the number of iterations required for intrusion detection.

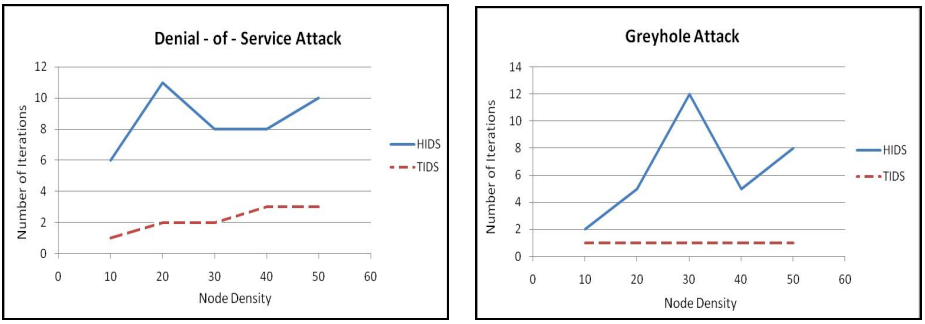

Fig. 3. Performance of TIDS and HIDS with variation in Node Density

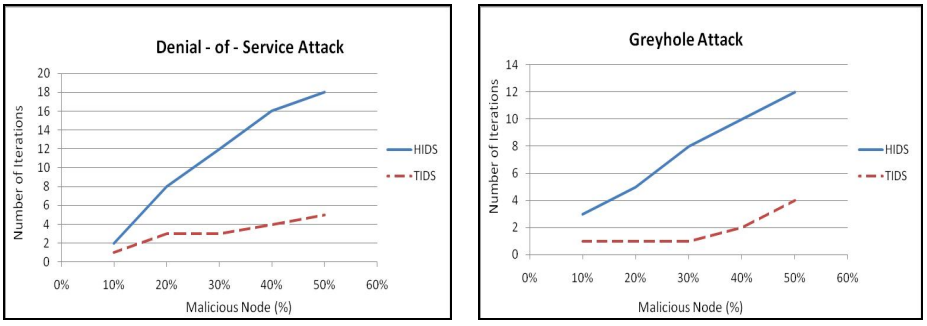

Fig. 4. Performance of TIDS and HIDS with variation in % of Malicious Nodes

The next set of data were taken with a fixed number of nodes (=40) and varying the percentage of malicious nodes. Both HIDS and TIDS performed reasonably well in terms of false negatives. None of the algorithms generated any false positives.

All results reflected the sensitivity of the newly proposed Trust model over the Honesty – based scheme proposed in HIDS. These results clearly indicate that TIDS is much more efficient in detecting malicious behavior.

5 Conclusion

In this paper a new trust based IDS has been proposed and evaluated against similar collaborative trust based IDS of recent past. In fact, the proposed TIDS not only detects intrusion, but also proactively responds in finding trusted routes based on this

detection. Thus, TIDS exceeds beyond just being an IDS and works more as an Intrusion Response System. The proposed methodology may be of extended for Wireless Mesh Networks and Sensor Networks. QoS management may be done efficiently using the proposed Trust model. Also, Trust based routing can be deployed in wireless networks to reduce the chances of possible intrusions in the first place. There is scope for implementation in the domain of MANETs where the proposed algorithm can be simulated by varying mobility of the nodes.

Acknowledgement. We would like to thank the Advanced Technology Cell, DRDO Cell for approving our work as a Defense – related project. The contingency and Fellowship provided by the Advanced Technology Cell has played a vital role in the completion and publication of this work.

References

1. Liang, Z., Shi, W.: PET: A PErsonalized Trust Model with Reputation and Risk Evaluation for P2P Resource Sharing. In: Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS 2005) - Track 7, vol. 7, p. 201b (2005)
2. Cho, J.H., Swami, A., Chen, I.R.: Modeling and Analysis of Trust Management with Trust Chain Optimization in Mobile Ad Hoc Networks. In: Proc. of the 2009 International Conference on Computational Science and Engineering, vol. 02, pp. 641–650 (2009)
3. Xiong, L., Liu, L.: Building Trust in Decentralized Peer-to-Peer Electronic Communities. In: Proc. 5th Int'l Conf. Electronic Commerce Research, ICECR-5 (2002)
4. Wang, W.G., Mokhta, M., Linda, M.: C-index: trust depth, trust breadth, and a collective trust measurement. In: Proceedings of the Hypertext 2008 Workshop on Collaboration and Collective Intelligence, pp. 13–16 (2008)
5. Luo, J., Liu, H.X., Fan, M.Y.: A trust model based on fuzzy recommendation for mobile ad-hoc networks. Computer Networks 53(14), 2396–2407
6. Sil, P., Chaki, R., Chaki, N.: HIDS: Honesty-rate based collaborative Intrusion Detection System for Mobile Ad-Hoc Networks. In: Proc. of 7th IEEE International Conference on Computer Information Systems and Industrial Management Applications, CISIM (2008)