Mobile Information Systems 2 (2006) 169–191
IOS Press

An agent based intrusion detection model for mobile ad hoc networks

B.M. Reshmi[a], S.S. Manvi[b] and Bhagyavati[c]
[a]*Department of Information Science and Engineering, Basaveshwara Engineering College, Bagalkot, India*
[b]*Department of Electronics and Communication Engineering, Basaveshwara Engineering College, Bagalkot, 587102, India*
E-mail: breshmi@yahoo.com,sunil@protocol.ece.iisc.ernet.in
[c]*Department of Computer Science, Columbus State University, 4225 University Avenue, Columbus, Georgia 31907-5645, University System of Georgia*
E-mail: bhagyavati@colstate.edu

Abstract. Intrusion detection has over the last few years, assumed paramount importance within the broad realm of network security, more so in case of wireless mobile ad hoc networks. The inherently vulnerable characteristics of wireless mobile ad hoc networks make them susceptible to attacks in-spite of some security measures, and it may be too late before any counter action can take effect. As such, there is a need to complement traditional security mechanisms with efficient intrusion detection and response systems. This paper proposes an agent-based model to address the aspect of intrusion detection in cluster based mobile wireless ad hoc network environment.

The model comprises of a set of static and mobile agents, which are used to detect intrusions, respond to intrusions, and distribute selected and aggregated intrusion information to all other nodes in the network in an intelligent manner. The model is simulated to test its operation effectiveness by considering the performance parameters such as, detection rate, false positives, agent overheads, and intrusion information distribution time. Agent based approach facilitates flexible and adaptable security services. Also, it supports component based software engineering components such as maintainability, reachability, reusability, adaptability, flexibility, and customization.

1. Introduction

Mobile ad hoc networks (MANETs) are self-organizing network architectures in which a collection of mobile nodes with wireless interfaces may form a temporary network without the aid of any established infrastructure or centralized administration. The hosts in MANETs are free to move randomly and organize themselves arbitrarily; thereby causing dynamic network topology. This allows for greater mobility and dynamic allocation of node's structures [1]. Wireless ad hoc networks find application in military operations so that planes, tanks, and moving personnel can communicate. Rescue missions and emergency situations can also use such networks. Other examples include virtual class rooms and conferences where in people can set up a network on the fly through their laptops, PDA's, and other mobile devices.

Ad hoc networks are characterized by dynamic topology, infrastructure-less, limited resources in mobile nodes, variable capacity links, low security, bandwidth limitations and energy-constraints. One of the important issues that must attract researcher's attention is security. The security goals of MANET

include availability, integrity, authentication, confidendentaility, and non-repudiation [2]. In order to achieve the security goals, MANET needs the following security means: protecting routing mechanisms, protecting key management schemes and intrusion detection. This section presents intrusion detection mechanisms and related works.

1.1. Intrusion detection

Intrusion detection system (IDS) serves as an alarm mechanism for a computer system. It detects the security compromises of a computer system and then issues an alarm message to an entity, such as site security officer so that entity can take some actions against the intrusion. An IDS contains an audit data collector, which keeps track of the activities within the system, and a detector which analyzes the audit data and issues an output report to the site security officer. In discussion of IDS with respect to MANET, two concepts need to be distinguished: intrusion detection technique and intrusion detection architecture. IDS techniques [3] refer to the concepts such as *anomaly and misuse detection*. Anomaly detection defines and characterizes normal or acceptable behaviors of the system (e.g., CPU usage, job execution time, system calls). Behaviors that deviate from the expected normal behavior are considered intrusions. Misuse detection characterizes known methods to penetrate a system. These penetrations are characterized as a 'pattern' or a signature' that the IDS looks for. The pattern/signature might be a static string or a set of sequence of actions.

The intrusion detection architecture, however, deals with problems of a larger scope. In wireless networks, however, it is very difficult for a node to make decision just based on locally collected data. Nodes must collaborate or exchange data at least in making an intrusion decision. Therefore an IDS architecture defines the roles of different nodes and the way they communicate.

Ad hoc networks are exposed to many possible attacks. We can classify these attacks into two kinds [4]: *Passive and Active attacks*. In passive attacks, attackers don't disrupt the operation of a protocol but only attempt to discover valuable information by listening to the traffic. Airwave monitoring can be used to deal with passive attacks. An active attack can mainly be of following types: black hole attacks, wormhole, routing tables overflow, sleep deprivation, location disclosure, denial of service and impersonation attacks. While passive attacks are rarely detectable, active ones can often be detected. It is expected that an ideal IDS is likely to support the following requirements [5].

- The IDS should not introduce a new weakness in the MANET. That is, the IDS itself should not make a node weaker.
- IDS should run continuously and remain transparent to the system and users.
- The IDS should use as little system resources as possible to detect and prevent intrusions.
- It must be fault-tolerant in the sense that it must be able to recover to the previous state, and resume the operations before the crash.
- Apart from detecting and responding to intrusions, IDS should also resist subversion. It should monitor itself and detect if it has been compromised.
- IDS should not only detect but also respond to detected intrusions, preferably without human intervention.
- Accuracy of the IDS is another major factor in MANETs. Fewer false positives and false negatives are desired.
- It should inter-operate with other intrusion detection systems [24] to collaboratively detect intrusions.

1.2. Related works

This section presents relevant works related to intrusion detection in MANET. (1) Zhang and Lee [6] describe a distributed and cooperative intrusion detection model where every node in the network participates in intrusion detection and response. (2) Bhargava et al. [7] proposed an intrusion detection and response model to enhance security in ad hoc on demand vector routing protocol. (3) TIARA [8] is a set of design techniques that strengthen MANETs against denial of service attacks. The TIARA mechanisms limit the damage sustained by MANETs from intrusion attacks and allow network operation to continue at an acceptable level during such attacks. (4) The LIDS [9] is distributed in nature and utilizes mobile agents on each of the nodes of the ad hoc networks. (5) Kachirski and Guha [10] have proposed a distributed intrusion detection system for ad hoc wireless networks based on mobile agent technology. By efficiently merging audit data from multiple network sensors, their bandwidth-conscious scheme analyzes the entire ad hoc wireless network for intrusions at multiple levels, tries to inhibit intrusion attempts, and provides lightweight low-overhead mechanism based on mobile agent concept. (6) A distributed IDS [11] has been proposed at Mississippi State University in which each node on the network has an IDS agent running on it. (7) Yi-an Huang and Wenkee Lee [19] have proposed a cluster based IDS; the member nodes in the cluster usually pass some local security information to the cluster head. Cluster head derives intrusion information independently by using the collected information.

In the field of computer security, one of the most damaging attacks is masquerading. A masquerade attack in which one user impersonates another is among the most serious forms of computer abuse [12]. Masquerade detection falls under the cover of anomaly detection. Detecting anomalous behavior can be viewed as a binary valued classification problem in which measurements of system activity such as system log files, resource usage, command traces, and audit trails are used to produce a classification of system's state as normal or abnormal. The problem of intrusion detection is inherently statistical because it is data driven.

Related works on masquerade detection includes the following: (1) Du Mouchel [13] proposed a Bayes 1-Step Markov detector based on single step transitions from one command to the next. The detector determines whether or not observed transition probabilities are consistent with historical probabilities. (2) The uniqueness of the approach [14] given by Schonlau and Theus is based on ideas about command frequencies. (3) The sequence match [15] approach presented by Lane and Broadley, computes a similarity matches between the most recent commands, user commands and a user profile. (4) The compression method [16] by Karr and Schonlau is based on the idea of compression approach.

We observe from the literature that none of the existing schemes comprehensively deal with the distributed intrusion detection in cluster based mobile ad hoc network by considering both host based detection technique and distributed intrusion detection. Also, the traditional schemes lack extensibility, customizability, software reuse, maintainability and flexibility. Agent technology seems to provide a solution to deal with these issues.

1.3. Proposed work

The proposed intrusion detection model differs from other mentioned works in the following aspects: 1) combination of distributed IDS and clustering, 2) statistical method for detecting masquerading attacks based on user activity patterns [21], 3) distribute the selected and aggregated masquerading information to all the clusters in an optimal way, 4) uses a set of static (manager agent and host intrusion detection agent (HIDA), cluster head manager agent) and mobile agents (distributing mobile agents) to detect masquerading in the cluster based ad-hoc networks. A mobile agent is a program that can migrate

from one host to another in a network of heterogeneous computer systems and fulfill a task specified by its owner. Each node in the cluster contains a static agent, HIDA, which is responsible for detecting masquerading (a security attack in which an intruder assumes the identity of a legitimate user).

The HIDA uses a statistical approach (anomaly detection method) to measure similarity between sequences of activities (activity patterns) produced by a potential intruder and the user signature, which is a sequence of activities collected from a legitimate user. If HIDA detects any intrusion in the audit data (sequence of user's activity patterns), it informs the manager agent in the node, which in turn informs the cluster head manager agent. The cluster head manager agent responds to the attack and launches a distributing mobile agent to propagate alert information to all other cluster head's manager agent in the ad hoc network.

1.4. Organization of the paper

The remainder of the paper is organized as follows. Section 2 describes the proposed intrusion detection model. The simulation model and the procedure are presented in Section 3. Section 4 discusses the simulation results. Sections 5 and 6 presents benefits of using agents and conclusions, respectively.

2. Proposed intrusion detection model

In this section, agent technology and the proposed model for the intrusion detection are presented. The model assumes that an agent platform is available in the mobile nodes of an ad-hoc network. However, if an agent platform is unavailable, the agents communicate by traditional message exchange mechanisms such as message passing and remote procedure calls. We assume that the mobile nodes automatically form clusters due to their limitations in the range of communication and elect their cluster heads. Also, model assumes that agents carry the data distribution by reserving some fraction of bandwidth at regular intervals or as and when required, i.e., priority is given to intrusion information distribution agents.

2.1. Agent technology

Agents are the autonomous or semi autonomous programs situated within an environment, which sense the environment and act upon it to achieve the goals. Agents have some special properties such as mandatory and orthogonal properties [17]. These properties make the agents different from the standard software. Mandatory properties are: autonomy, reactive, proactive, and temporally continuous. Here, we give the properties with respect to ad hoc network environment.

- Autonomy. Agents can function without any regular initiation from the user or processes. They can start working once initiated by a user or a process. Some of the activities where this feature can be observed are: monitor the battery life, power requirements to neighbors, reliable neighbors, discover routes in anticipation to link breaks, checking intruders, studying legitimate user behavior patterns, etc. This allows them to operate independently.
- Reactivity to the environment. With this, the agent must be able to react to changes in its environment such as changes in user behavior, change in neighbors of a node, etc.
- Pro active and goal oriented. Agents anticipate the changes in the MANET environment and take appropriate decisions. A key factor with agents is that they typically have a single task to complete, such as, monitoring user behavior, monitoring for a user login, etc. This gives the agent a small foot- print and also make them easier to test, and are more robust.

– Temporal continuity. An agent continuously runs in order to test, diagnose, and control the network environment.

The orthogonal properties are: communicative, mobile, learning, and believable.

– Communicative. Agents communicate with other agents, processes, and users to gather some information about the network environment such as dynamic topology, unreliable nodes, intrusions, resource availability, etc. and the decisions taken on the network.
– Learning. It is often not possible to upgrade agents on a regular basis once they are deployed. Thus as much as possible, an agent should try and learn from its experience or by other agents. Machine learning techniques such as neural networks and reinforcement learning can also be employed in agents.
– Mobile. An agent can be mobile, i.e., move from one place to another facilitating asynchronous communication between the nodes in MANET.

A mobile agent differs from static agents in terms of mobile property. It is an itinerant agent that is dispatched from a source node that migrates from one host to another in the heterogeneous network and executes at the remote host until it accomplishes its task. The agent may contain the program, data, and execution state information.

Mobile agents and wireless networks are two cutting edge technologies that will provide enhancements for increased connectivity and communication. Mobile agents are asynchronous, i.e., they do not need permanent network connectivity, which is more suitable in case of wireless networks since wireless channels are less reliable. Mobile agents can interact with environment by communicating with other static and mobile agents in the network.

Mobile agents imply numerous benefits [18]:

– Reduce the network load. Client server or peer-to-peer architecture often uses lot bandwidth resources, especially when accessing servers, which tend to become bottlenecks. When large volumes of data are stored on servers, it is more efficient to move the process to the location of data, instead of data itself.
– Reduce latency. When more number of request-responses as used in client-server architecture is required to complete a given task, latency will be more. With code sent to server, request-responses taking place locally; latency will be reduced since interactions are done locally rather than over a network.
– Embedded protocols. They can embed some protocols, thus allowing for easier updates. A good example of this would be in an update to an application program, which uses mobile agents. With this all the application agents could be recalled to a central source for up gradation and redeployed.
– They interact with their environment and adapt themselves.
– Move autonomously. This autonomy along with platform and system independence make them ideal for building reliable and robust distributed applications and can thus deal with the environment reacting dynamically to changes.

A key element of mobile agents is the development system, which should provide a platform for the services required for mobile agents. Different agent programming platforms are available. An agent platform comprises of an agent server, agents, security manager, agent transport mechanism, and an execution environment. The services provided by an agent platform are: agent creation, agent mobility, security, persistence, agent execution, inter-agent communication, and messaging.

Some existing agent platforms are IBM aglets, odyssey, grass hoper, and voyager [22], which suits wired network environment. A mobile code platform using J2ME for ad hoc networks is given in [23].

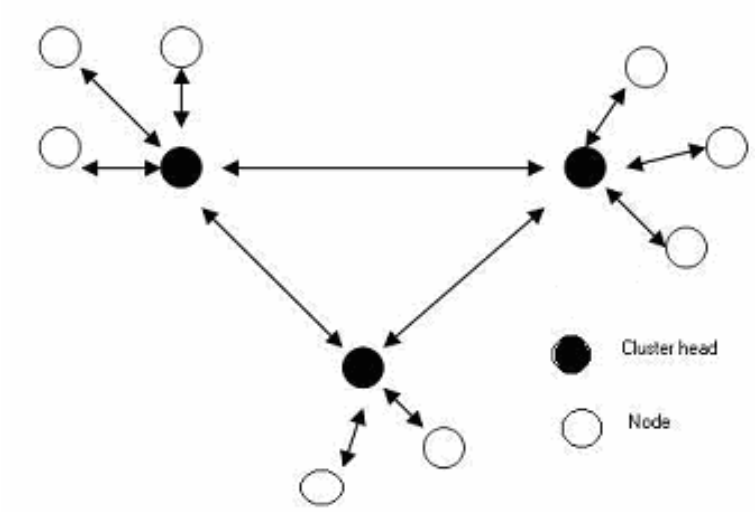


Fig. 1. Hierarchical network infrastructure.

However, standard platform is yet to be developed. Mobile agent code is platform independent. As a result, it can execute at any remote host in the heterogeneous network environment. Agents communicate and cooperate with each other in order to achieve their goals. An agent can update its knowledge base while interacting with other agents during its travel over the network. Inter-agent communication can be achieved by message passing, remote procedure call, or common knowledge base called as blackboard.

Mobile agents have advantages that make them very useful in a variety of applications, such as network management, distributed computing, e-commerce, grid computing, data mining, information management, etc. Agents have many characteristics that enable them to enhance intrusion detection technology. Secured routing of mobile agents to perform a given task is an essential feature to perform intrusion detection. The work given in [25] presents a scheme for secured routing of mobile agents in distributed systems. Mobility is obviously one of the most important capabilities, and we can certainly benefit from it. Mobile agents can avoid having to transfer intrusion detection data located at a collection point, back to a central repository and allows distributed detection of intruders. Mobile agents can also be viewed as a collection of guards. Security companies do not want to incur the expense of posting a guard in every hall of a building. Instead, they have a guard walk through each hall periodically checking for intrusions. Likewise, Mobile agents enable one to periodically check hosts for security problems without having to install checking software on every host.

2.2. Network environment

Wireless ad hoc networks may be configured in basically two ways, either a flat network infrastructure or a multi-layered network infrastructure (cluster based mobile ad hoc network). The proposed intrusion detection model is used for the multi–layered infrastructure depicted in Fig. 1, where nodes within transmission range are organized into a cluster, and elect a Cluster Head (CH) node to facilitate communication [20]. The CH nodes are more powerful devices with better resources and they form a virtual backbone of the network.

Depending on the protocol, intermediate gateway nodes may relay packets between CH nodes. Therefore, the major part of the processing in detecting intrusion can be done on the CH nodes. That is, each

CH node is responsible for the cluster of devices that are communicating with it. This affects the design and implementation of an intrusion detection system in a big way as sophisticated forms of intrusion detection can be used with the CH acting as a central unit and serving as a certification authority for providing trust among the devices.

2.3. An agent based intrusion detection by using activity patterns

The proposed model uses a set of static and mobile agents that have the ability to support asynchronous communication and flexible service processing. The proposed model consists of intrusion detection agency at every node of a cluster and the CH in MANET. Agency at every node in the MANET captures data locally, makes decision about intrusions and informs cluster head agency. Later, cluster head agency uses mobile agents to selectively aggregate the intrusion information and distribute to the visited cluster head nodes. Cluster head nodes in turn distribute the information to all the nodes within it. The intrusion detection agency at every node comprises of a manager agent, an intrusion detection agent and a blackboard for inter-agent communication. The agency in CH comprises of CH manager agent, intrusion detection agent, distributing mobile agents and a blackboard for inter-agent communication.

The reasons supporting intelligent working of the proposed model are as follows. 1) HIDA has learning capability, i.e., it updates training data of a legitimate user at regular intervals and uses it for detecting intrusions even though users move and the network topology changes. 2) Mobile agents aggregate the selected masquerading information (only intrusion activity patterns and the user identification) from the visited clusters while distributing the intrusion information to all other nodes on its journey. 3) Detection agent selects a threshold to detect intrusions based on scores computed for different activity patterns for a user.

2.3.1. Node agency

The functional architecture of the intrusion detection agency used in each node of a cluster other than CH is given in Fig. 2.

- *Blackboard:* Blackboard is a knowledge base that can be read and updated by the agents. The knowledge base facilitates inter-agent communication. The data stored in the knowledge base are as follows: user identification, historical audit data for a user (training data or activity patterns), user normal profile (in terms of statistical values), test data for detecting intrusions, local table (contains response generated by host intrusion detection agent for an intrusion, misuse activity patterns) global table (contains responses generated by other nodes to intrusion, node id, CH id, information about neighbors for routing, signal strengths, etc., for the nodes).
- *Manager agent* (MA): It is a static agent that creates a host intrusion detection agent and a blackboard. This agent coordinates all the actions of agents in the agency. Manager agent is responsible for sending intrusion information (if it is new) detected by the host intrusion detection agent to the CH manager agent for distributing this information to all other nodes in the network. This agent has following properties: autonomous, proactive, goal oriented, temporal continuity and communicative.
- *Host intrusion detection agent* (HIDA): This is a static agent consisting of three important modules: (1) audit data collector, a module for collecting audit data, (2) profile builder, a module for constructing normal profile for each user, and (3) local intrusion detection module, responsible for detecting intrusions (masquerading). This agent has the following properties: autonomous, proactive, goal oriented, temporal continuity, learning and communicative. Figure 3 depicts the various elements of HIDA.

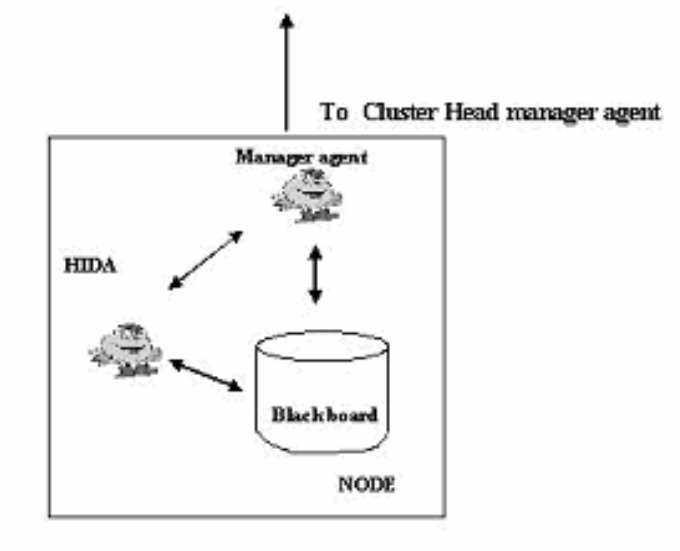


Fig. 2. Intrusion detection agency at a node (other than cluster head) in MANET.

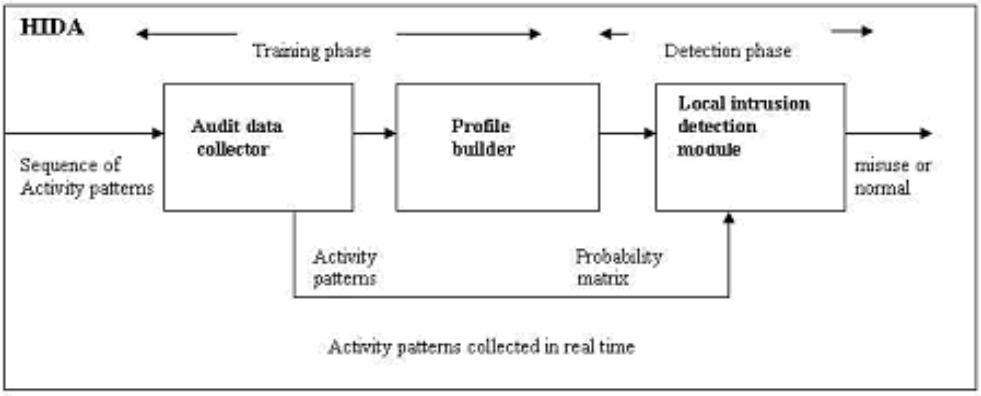


Fig. 3. Elements of HIDA.

The elements of HIDA are explained below.

– *Audit data collector:* In order to recognize anomalous behavior, it creates a user profile to characterize normal behavior. To create a user profile our approach learns characteristic sequences of actions generated by users. The underlying hypothesis is that a user responds in a similar manner to similar situations leading to repeated sequences of actions. It is the differences in characteristic sequences that we attempt to differentiate a valid user from an intruder masquerading as that user. An audit trail is maintained for a variety of user activity types, for example, logging, operating system commands, database system interrogations and updates and the details of user interactions with specialized programs.

Audit data collector module is responsible for collecting characteristic sequence of actions (i.e., activity patterns or sequence of commands executed), an ordered, fixed or variable set of temporarily adjacent

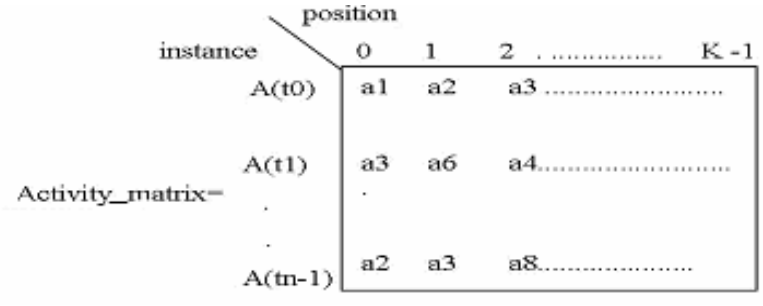


Fig. 4. Activity matrix.

actions for each user at different instances on regular basis. This module translates the raw data stream of activity traces into a suitable format for storage and comparison. This translation suppresses activity arguments and preserves only the activity names.

To define activity pattern for each user, first, we define an activity set, that contains all possible activities executed by the users (for example, login, mail,. . .) which is common for all the users i.e., $A = \{a0, a1, a2, a3, \ldots, a_{K-1}\}$, where, each element of the set represents a single command executed by a user and, K= total number of activities. Audit data collector collects different activity patterns for each user at different instances in training phase on regular basis and updates blackboard. Activity patterns collected for 'n' instances at different times, $T = \{t0, t1, \ldots, tn-1\}$, is expressed as follows.

$A(t0) = \{a1, a2, a3, a5, a6, a7, a8., a7, a9 \ldots, a5\}$,

$A(t1) = \{a3, a6, a4, a9, a6, a5, a3, a6, a4, a3 \ldots, a2\}$,

.....

.....

$A(tn-1) = \{a2, a3, a8, ..., a10\}$.

Activity patterns collected for 'n' instances are represented as an activity matrix, in which each row represents an activity pattern instance and each column represents the position of an activity in an activity pattern as given in Fig. 4.

– *Profile builder* (*training phase*)*:* Masquerade detection (anomaly detection) consists of two phases: training and detection phase. In training phase, normal profile for each user is constructed on regular basis as said above. Whereas in detection phase, anomaly detector monitors current user session to detect intrusions. The normal profile is used to learn the user normal behavior. In the training phase, profile builder computes a probability matrix for each user, which represents a user normal profile, based on activity patterns collected by audit data collector, by using statistical method. Probability matrix represents a normal behavior of a user in terms of probability of executing each activity in different positions. The normal profile in the form of probability matrix is stored in the blackboard. This profile is then used to detect intrusions in the detection phase. The probability matrix is as shown in Fig. 5.

In the probability matrix, row represents activity identity and column represents probability of an activity execution in that position (column number), where Z = max number of sequence of activities traced in each instance, and K = maximum number of activities identified in the system.

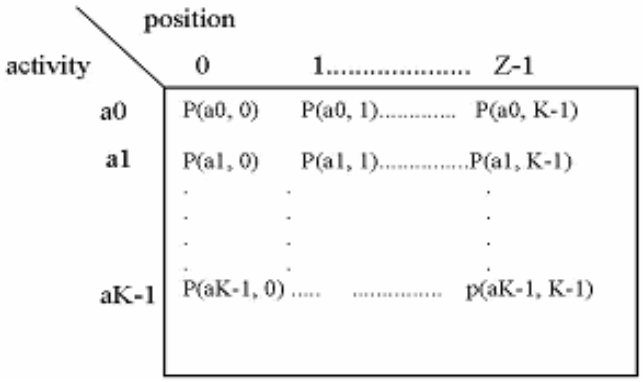


Fig. 5. Probability Matrix.

For example, $P(a0, 0)$ represents probability of executing an activity 0 ($a0$), in the position 0 (i.e., in the first position), $P(a1, 0)$, probability of executing an activity 1 ($a1$), in the position 0. Computation of probability is done as follows which is given for $P(a0, 0)$.

$$P(a0, 0) = \frac{\text{frequency of executing an activity 0 (a0) in position 0}}{n}$$

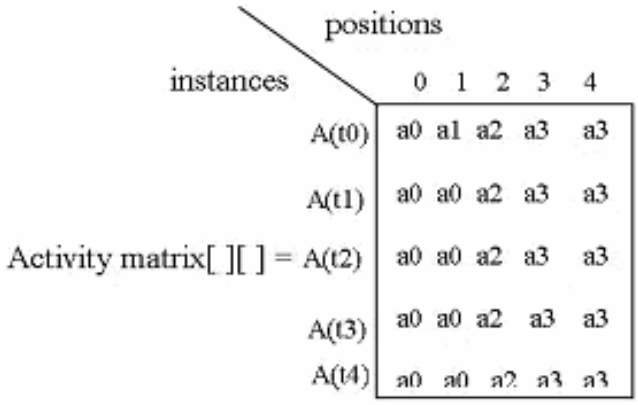
where, $n =$ Total number of instances in the training data.

– *Local intrusion detection module* (*detection phase*)*:* Once a user profile is formed, the basic action of the local intrusion detection system is to compare the incoming sequences to the historical data and form an opinion as to whether or not they both represent the legitimate user. The dynamic behavioral patterns are updated at regular intervals for a legitimate user as follows. After every 50 (for example) test patterns for a given user (i.e., after patterns 50, 100, 150 and 200), we consider updating the training data. When detection alarm is not raised in the recent 200 patterns for a given user, training data set is replaced by the most recent 200 activity patterns and the new probability matrix for the user is recomputed. If one or more alarms were raised, the previous training data and the probability matrix remain in place.

We assign different threshold to all users based on their scores calculated for each activity pattern in the test data. For each activity pattern, local intrusion detection module computes a score, if this score exceeds some threshold value, it declares, that activity pattern is a misuse pattern. To compute a score for each activity pattern in observation, it considers each activity and its position. For each activity in the activity pattern, it starts searching a probability matrix to know a highest probability of executing an activity in that position and also the probability of executing an activity under consideration in the same position and the difference between these two results a deviation value for an activity. That is, deviation is computed by subtracting the elements of the probability matrix determined from the test data from the probability matrix generated in learning phase. Product of all the deviation values gives a score for the single activity pattern. We compute score for each activity pattern in the test data, and consider three different threshold values i.e. (1) minimum, (2) average and, (3) maximum of computed scores to observe the detection rate.

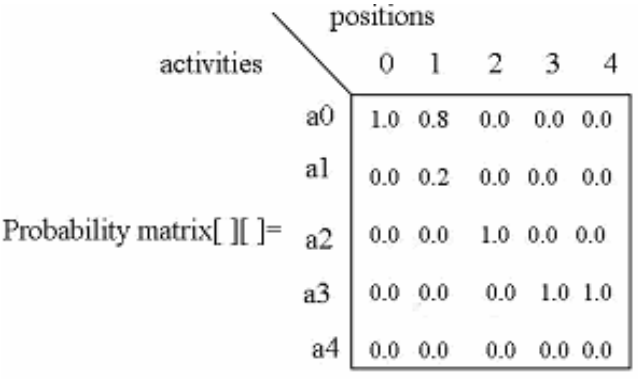
Consider an example, to illustrate the intrusion detection, let us say, Activity set is: $\{a0, a1, a2, a3, a4\}$. Activity patterns collected at five instances (training data) for a user 'x', are as follows.

$A(t0) = \{a0, a1, a2, a3, a3\}$,



Activity matrix[ ][ ] =	positions / instances	0	1	2	3	4
	A(t0)	a0	a1	a2	a3	a3
	A(t1)	a0	a0	a2	a3	a3
	A(t2)	a0	a0	a2	a3	a3
	A(t3)	a0	a0	a2	a3	a3
	A(t4)	a0	a0	a2	a3	a3

Fig. 6. Activity matrix for example activity pattern.



Probability matrix[ ][ ]=	positions / activities	0	1	2	3	4
	a0	1.0	0.8	0.0	0.0	0.0
	a1	0.0	0.2	0.0	0.0	0.0
	a2	0.0	0.0	1.0	0.0	0.0
	a3	0.0	0.0	0.0	1.0	1.0
	a4	0.0	0.0	0.0	0.0	0.0

Fig. 7. Probability matrix for example activity pattern.

$A(t1) = \{a0, a0, a2, a3, a3\},$

$A(t2) = \{a0, a0, a2, a3, a3\},$

$A(t3) = \{a0, a0, a2, a3, a3\},$

$A(t4) = \{a0, a0, a2, a3, a3\},$

The activity matrix and probability matrix for user 'x' is as given in Figs 6 and 7.

To illustrate the elements of probability matrix, let us consider one of the computations, $P(a0, 0)$. From the instances, we see that a0 occurs in the first position of all the 5 instances, hence $P(a0, 0) = 5/5 = 1.0$.

Consider each column in the probability matrix. Highest probability value for an activity a0 executing in the position 0, i.e., in the column 0 is 1.0; for an activity a0 executing in the position 1, i.e., in the column 1 is 0.8; for an activity a2 executing in the position 2, i.e., in the column 2 is 1.0; for an activity a3 executing in the position 3, i.e., in the column 3 is 1.0; for an activity a3 executing in the position 4, i.e., in the column 4 is 1.0.

Consider a test data for user 'x', at three different instances to check for intrusions as given below.

$A(t0) = \{a0, a0, a2, a3, a3\},$

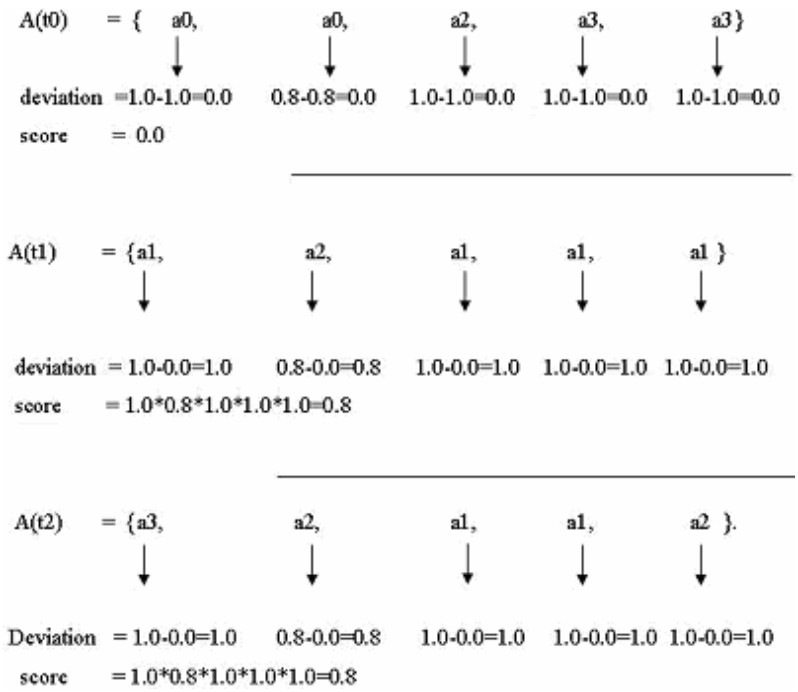


Fig. 8. Deviation value and scores for example activity patterns.

$A(t1) = \{a1, a2, a1, a1, a1\}$,

$A(t2) = \{a3, a2, a1, a1, a2\}$.

The deviation value for each activity in the activity patterns and score for each activity pattern in different instances is calculated as given in Fig. 8.

Consider threshold score value to be maximum of three scores, which is 0.8. The first activity pattern is not a misuse activity pattern. Since score $<$ threshold. The second activity pattern is a misuse activity pattern. Since score $=$ threshold. The third activity pattern is a misuse activity pattern. Since score $=$ threshold. The third activity pattern is not seen in the user's historical data (training data or normal user profile), i.e., anything not seen in the historical data represents a different user (intruder or masquerader).

Once it detects masquerading, the local table of the blackboard is updated with the following information in the format, $<$node-id, user-id (intruder id), misuse activity patterns, response, and timestamp$>$ and it informs the same to the manager agent. The manager agent in the node sends masquerade information to the CH's manager agent for distributing the same to the other nodes in the ad hoc network.

2.3.2. CH agency

The functional architecture of the Intrusion detection model used in each CH of the cluster is given in Fig. 9.

Now we describe each component of the model.

– *Black board:* Blackboard is a knowledge base that can be read and updated by the agents. The knowledge base facilitates inter-agent communication. The information stored in the blackboard are: local table (contains masquerade information collected from the nodes within the cluster in the format $<$node-id, user-id, misuse activity patterns, timestamp$>$), and global table (contains

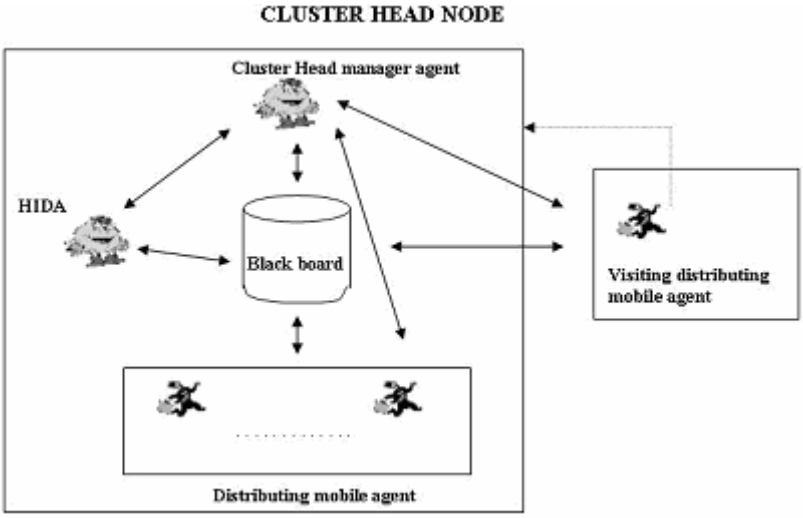


Fig. 9. Intrusion agency model at a cluster head node.

masquerade information collected from other cluster heads in the format: <cluster head id, user id, node id, misuse activity patterns, response, time stamp>, and neighborhood information (node and cluster addresses) for sending mobile agents).

- *Cluster Head manager agent* (CHMA): It is a static agent that collects masquerade information from the nodes within the cluster periodically or as and when required. It broadcasts this information within the cluster and also it distributes this information by launching distributing mobile agents by encapsulating masquerade information (alert message) to other nodes in the MANET. This agent has the following properties: autonomous, proactive, goal oriented, temporal continuity, and communicative.
- *Host intrusion detection agent* (HIDA): This agent is same as the one used in the intrusion agency framework at the node.
- *Distributing mobile agent* (DMA): These are the mobile agents that migrate from one CH node to another CH node in a loop free manner (visits a CH only once) to distribute masquerading information, collected from cluster nodes within the cluster. This agent has the following properties: autonomous, proactive, goal oriented, temporal continuity, mobile and communicative. Mobile agents are cloned from one CH node to its neighbor CH nodes. Agent cloning is a process of generating a number of similar copies of the agent. This agent provides information to the visited CH nodes about sender identification (CH node id), node id, intruder identification, misuse activity patterns, and time stamp and updates knowledge base at the visited CH node. Also it collects recent intrusions (if it is new type of intrusion other the earlier distributed intrusions) happened in that node by interacting with the CH manager agent through knowledge base, aggregates the intrusion information and distributes the information on its way by using cloned agents. The agent disposes itself in case of unavailability of unvisited neighbors.

The proposed model functions in two different phases: 1) Local intrusion detection in each cluster node and 2) distribution of masquerade information from one CH node to another CH node in the ad hoc network. When HIDA agent detects intrusion locally, it informs the CH manager agent. Periodically, CH manager agent distributes locally detected intrusions to all other CH nodes across the network, by using

distributing mobile agents. Once the CH node collects masquerade information from other nodes in the network, it broadcasts the same to all the nodes within that cluster. Now cluster nodes have information about masquerading, they can be aware of masqueraders. Algorithm given below presents pseudo code of the functioning of the model.

The proposed intrusion detection model differs from other mentioned works in the following aspects: 1) combination of distributed IDS and clustering, 2) statistical method for detecting masquerading attacks based on user activity patterns, 3) distribute the selected and aggregated masquerading information to all the clusters in an optimal way, 4) uses a set of static (manager agent and host intrusion detection agent (HIDA), cluster head manager agent) and mobile agents (distributing mobile agents) to detect masquerading in the cluster based ad-hoc networks, 5) dynamic behavioral patterns are considered by updating at regular intervals, 6) detection of new attacks is possible because of behavioral learning capability of the detection agency.

2.3.3. Algorithms

This section presents the algorithm for the proposed intrusion detection model using agents.

Algorithm: Agent based Intrusion detection

{ *Nomenclature*: M = the number of clusters, N = the number of mobile nodes in each cluster, u_{ji} = user of node j in cluster i, CH_i = Cluster Head node of cluster i, DMA_i = distributing mobile agent generated by CH_i, $HIDA_{ji}$ = Host intrusion detection agent at node j of cluster i, MA_{ji} = manager agent at node j of cluster i, $CHMA_i$ = cluster head manager agent of cluster i, th_{ji} = threshold value for user of node j in cluster i.}

Begin

1. For i = 1 to M do // Training phase of Anomaly detection
   Begin
2. For j = 1 to N do
   Begin
   - MA_{ji} triggers the $HIDA_{ji}$ and creates blackboard at each node of the cluster.
   - For each u_{ji}, Audit data collector module of $HIDA_{ji}$ collects activity patterns at different instances in regular basis, constructs an activity matrix (historical data or training data), and a profile builder computes a probability matrix (normal profile of a user). This information is updated into the black board at each node of the cluster.
   - Set th_{ji}.

   End // End of " for j " loop

   End // End of " for i " loop
3. For i = 1 to M do //Detection phase
   Begin

   For $j = 1$ to N do

   Begin

   For each u_{ji}
   - Audit data collector module of $HIDA_{ji}$ monitors test data (consisting of normal activity patterns and misuse activity patterns) of a user in real time.

- Local intrusion detection module of $HIDA_{ji}$ computes a score based on the probability values available in the probability matrix for each activity pattern in the test data.
- For each activity pattern, if the score exceeds threshold th_{ji}, that activity pattern is declared as a misuse activity pattern otherwise normal.
- If local intrusion detection module detects any masquerading, this information is updated into local table of blackboard along with response taken and the MA_{ji} is informed about this.
- The MA_{ji} sends masquerading information to the $CHMA_i$ of CH_i if it is a new intrusion other than the already distributed information.
- CHMA$_i$ updates masquerading information into its the global table of blackboard.

End //End of "for j" loop
End // End of " for i " loop

4. //distribution of masquerading information across the network
   - If any $CHMA_i$ finds masquerading information in Cluster i, it broadcasts this information within the cluster and it generates distributing mobile agent, DMA_i by encapsulating the following information: cluster head id, node id, user id, masquerader id, misuse activity patterns, response taken, timestamp.
   - DMA_i migrates from one CH_i to another CH_i for distributing masquerade information by using agent cloning process.
   - At a visited CH_i
     * DMA_i updates encapsulated information into the CH_i's blackboard.
     * DMA_i collects the any recent intrusions (if it is new) happened in that cluster by interacting with the $CHMA_i$ through the blackboard and aggregates with the available intrusion information and distributes on its way.
   - Each $CHMA_i$ broadcast masquerading information collected from other cluster head to all the nodes within that cluster.

   END
5. STOP

2.4. Example

We use a simplified example to demonstrate the proposed model. Consider a cluster based MANET consisting of two clusters 1 and 2 as shown in Fig. 10. Each cluster contains three nodes A, B, C, and D, E, F, respectively. C and D are cluster heads of cluster 1 and 2, respectively.

In this example, node B of cluster 1 is masqueraded. Masquerading is detected by the intrusion agency at node B and then the following sequence of operations takes place afterwards.

1. Manager agent of node B informs the CH manager agent (node C) about Masquerading.
2. CH manager agent of node C broadcast to node A that B is masqueraded.
3. CH manager agent distributes the same to other CH node i.e., D of cluster 2 by sending distributing mobile agent.
4. CH manager agent of node D broadcast the masquerading information to all the nodes in the cluster 2.
5. Distributing mobile agent disposes.

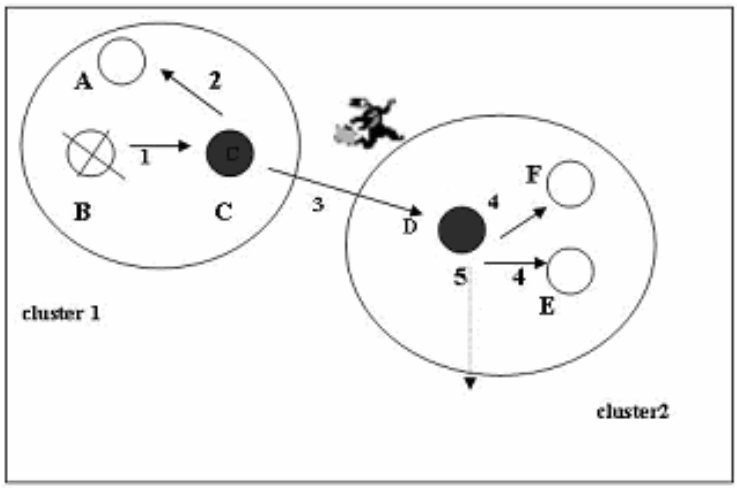


Fig. 10. An example model of agent based intrusion detection system.

3. Simulation model

The proposed model has been simulated in various network scenarios on a pentium-4 machine by using the C language. In this section we describe the network model and the simulation procedure.

3.1. Network model

An ad-hoc network consisting of '*num*' nodes is generated by using a random placement of the nodes within the area axb square meters (area is divided into grids of size 'g' square meters). The nodes are randomly arranged into c clusters, based on the communication range. The speed of movement of the individual nodes ranges from 'x' to 'y' meters/second. The number of nodes that move in the network is a fraction (f) of *num*. Each node starts from a random location and moves in any one of the eight directions: North, South, East, West, Northeast, Southeast, Northwest, and Southwest. Once it reaches the boundary, it moves in the opposite direction. An agent can travel at-most *maxhop* hops.

The signal range of each node is defined to be one block (grid) around it. Any two whose signal range overlaps are considered to be the neighbors. Each node maintains a list of its neighbors and its cluster head identification. Agent processing and migration time is uniformly distributed within the range $(m1, m2)$ milliseconds.

3.2. Intrusion model

We evaluate the proposed method using two types of intrusions, i.e., 1) intrusion detection considered at a host by using several masquerades on it, 2) intrusion detection in the network.

CASE I: *Intrusion detection at host*

Training and test data are generated for a user. A training data set (collection of legitimate activity patterns/sets) is randomly generated. An activity set defined for a user consists of K distinct activities. A random number generated between 0 and $K - 1$ in sequence for K times will form the sequence of activities for a user. Training data set for a user is generated for $n$instances. Test data set consists of both normal activity patterns already generated in training and misuse activity patterns. Test data set

contains p activity patterns, where some of the activity patterns (*np% of p*) are picked from training data set and the rest (*mp% of p*) are generated using different random number seed to create patterns other than normal user patterns, where $mp + np = 100\%$. Masqueraders use test data set to create intrusions. Number of masqueraders considered is *hitru.*

CASE II: *Intrusion detection in network*

We consider the model given above for a host and generate the intrusions by distributing the *nitru* masqueraders in a cluster-based network. Each masquerader can intrude in more than one node. Masquerader randomly selects the activity patterns from test data set for intruding. The node to be masqueraded is randomly selected among *num* nodes.

3.3. Simulation procedure

To illustrate some results of the simulation, we have taken num $= 25$, $c = 3$ to 5, $a = 500$ meters, $b = 700$ meters, $g = 15 \times 15$ metres, i.e., 225 grids and the size of each grid is (500/15) $\times$ (700/15) square meters, $x = 0$ meters/second, $y = 20$ meters/second, $f = 0.1$ to 0.3, $m1 = 100$ milliseconds, and $m2 = 200$ milliseconds, $K = 25$ activities, $n = 30$, 40, and 50 instances, $p = 10$ instances, mp is varied from 10 to 100%, hitru $= 5$, nitru $= 1$ to 10.

Begin

1. Generate an ad-hoc network with the given number of nodes.
2. Randomly generate training data (normal activity patterns collected at different instances) to train the system only on the normal data.
3. Randomly generate test data (contains normal activity patterns and misuse activity patterns) to evaluate the model.
4. Apply the proposed model to detect masquerading across the network.
5. Compute the performance of the system.

The performance parameters measured are as follows.

– *Detection rate:* It is a ratio of number of given activity patterns detected as intrusions (patterns include both intrusion and non-intrusion behaviors) by the system to the number of activity patterns given to the system.
– *False positive*: False positive is a non-intrusion activity pattern that the IDS has labeled as containing an intrusion.
– *Agent overheads:* Agent overhead is defined product of number of distributing mobile agents generated and the number of hops each agent travels for distributing aggregated intrusion information to all the nodes in the network.
– *Intrusion information distribution time:* It is defined as the time required for distributing intrusion information across the network.

4. Results

Figures 11, 12, 13 and Table 1 presents the results for *CASE I*. The detection rate ($n =$ number of instances in the training data $= 30$, 40 and 50 and the threshold (δ) value is minimum of the scores computed for each activity pattern for the test data) is observed, as 100% for all the intrusions as seen in Fig. 11. Even though the detection rate is 100%, it yields more number of false positives (false positive rate is very high).

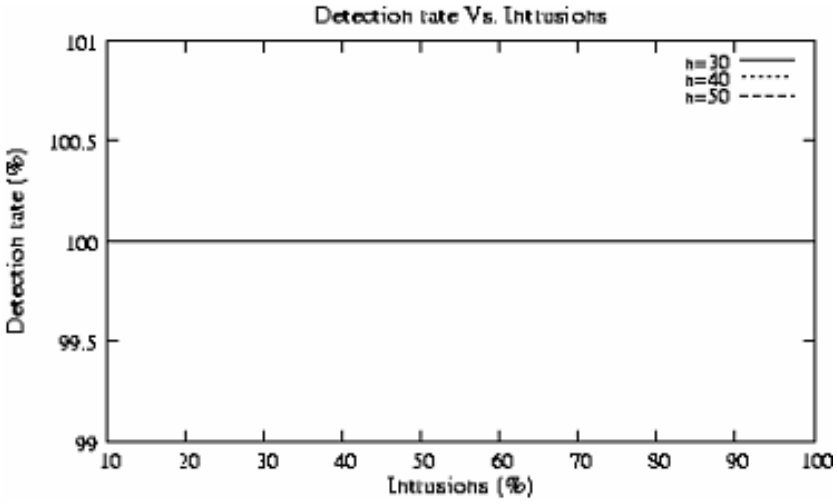


Fig. 11. Detection rate Vs. % of intrusions (δ = minimum of scores).

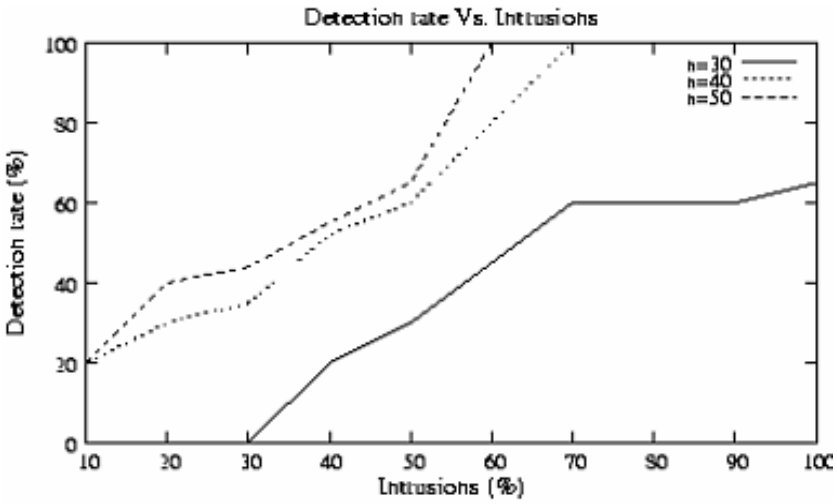


Fig. 12. Detection rate Vs. % of intrusions (δ = average of scores).

Figure 12 depicts that detection rate (n = number of instances = 30, 40 and 50 and the threshold value is average of the scores computed for each activity pattern in the test data) increases as the size of the training data set increases. It is also observed that the detection rate reaches 100% slowly.

From Fig. 13, we observe that the detection rate (n = number of instances = 30, 40 and 50 and the threshold (δ) value is maximum of the scores computed for each activity pattern) increases as the size of the training data set increases. It is also observed that it results in a few false positives. False positives are very much reduced with increase in training data set. Detection rate is better and yields lesser false positives as compared to Fig. 12.

Now, we present detection rates and false positives for different cases of threshold score values and instances for a fixed maximum number of intrusions (see Table 1).

For δ = minimum of scores it is observed that detection rate increases as the size of the training data

Table 1
Detection rate and false positives

n	Detection rate	False positives
Threshold = Minimum of scores		
50	100%	100%
80	100%	100%
100	100%	100%
Threshold = Average of scores		
50	40%	71.42%
80	80%	8.57%
100	80%	5.7%
Threshold = Maximum of Scores		
50	60%	28.57%
80	80%	2.75%
100	100%	1.65%

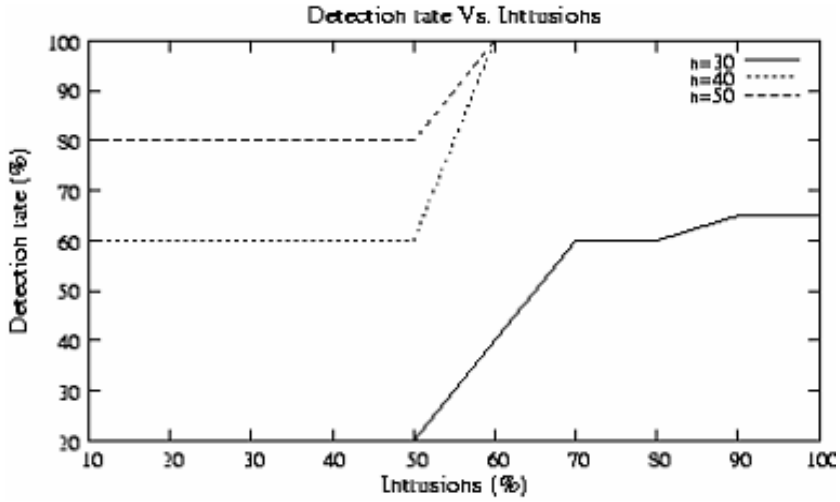


Fig. 13. Detection rate Vs. % of intrusions (δ = maximum of scores).

set (n) increases. But the false positive rate is very high, i.e., all detection done is false. False positives are equal to 100% when threshold is equal to minimum of scores due to following reason. Minimum of scores chooses the score value to be very low thus neglecting other higher score activity patterns, which leads to more number of false positives. It is observed that the detection rate increases as the size of the training data set increases for δ = average of scores and reaches 100% for $n = 100$ and results in reduced false positives. For δ = maximum of scores it is observed that detection rate increases as the size of the training set increases and reaches 100% for $n = 100$. It is also observed that false positive rate decreases with increase in size of the training data set but results in very few false positives as compared to the results observed for δ = average of scores.

We find from the results choosing δ = maximum of scores provides reduced false positives as compared to maximum of scores. It is better to consider threshold as maximum of scores.

We present detection rate and false positives for *CASE II* in Figs 14 and 15 with $n = 100$, and δ = maximum of scores, with different number of clusters. We observe that detection rates are around 80% to 85% and false positives are between 2% and 3.5%. Thus the system detects the intrusions with very less false positives.

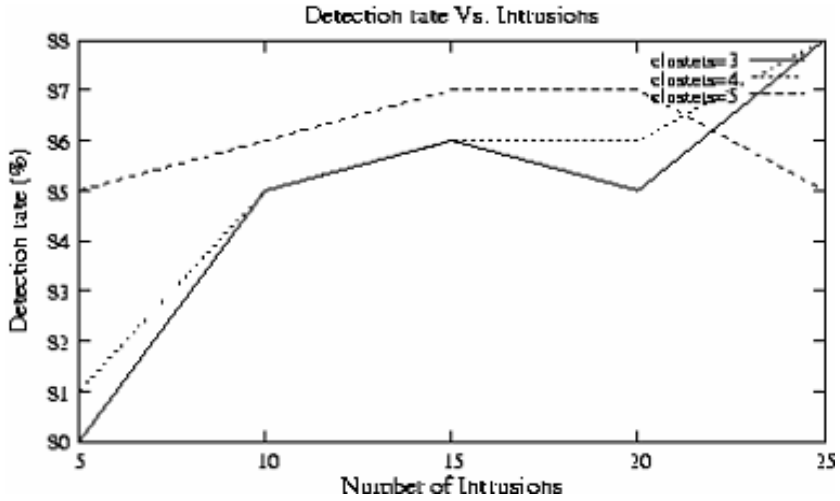


Fig. 14. Detection rate Vs. Number of Intrusions (for δ = maximum of scores).

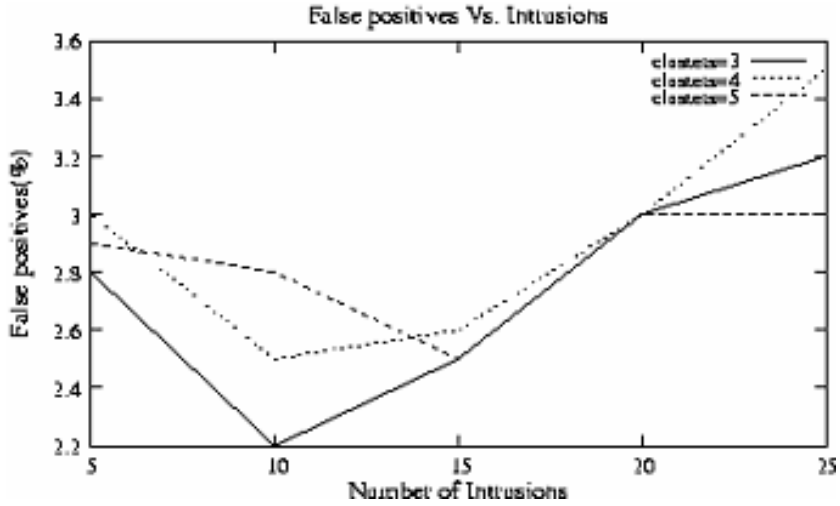


Fig. 15. False positives Vs. Number of Intrusions (δ = maximum of scores).

It is observed that the agent overhead increases with the number of intrusions and the number of clusters (see Fig. 16). This is mainly because agents have to travel more number of hops with increase in clusters.

We notice that intrusion information distribution time increases with increase in the number of intrusions and the number of clusters (see Fig. 17) because the agents have to roam to more number of nodes with increase in clusters.

5. Benefits with agents

We experience that the agent based intrusion detection model offers flexibility, scalability, efficiency, adaptability, software reusability, and maintainability. Even though it is difficult to quantify these

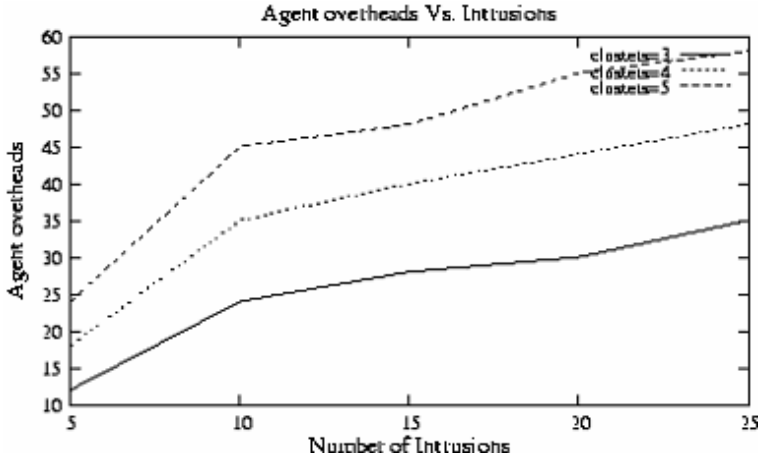


Fig. 16. Agent overheads Vs. Number of intrusions.

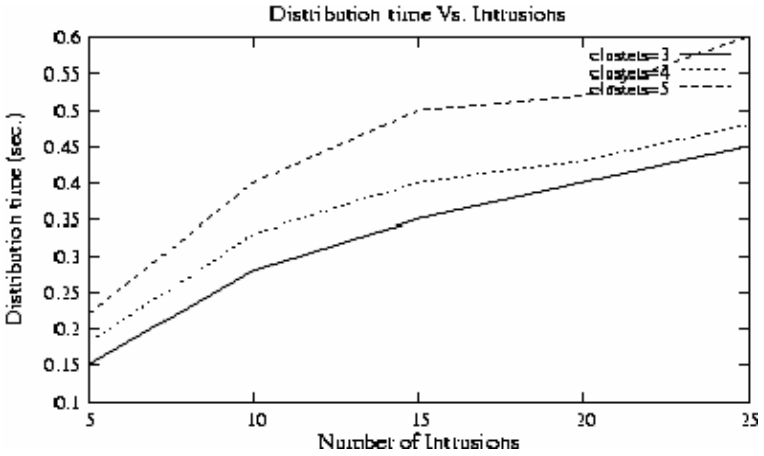


Fig. 17. Distribution time Vs. Number of intrusions.

features, below we explain how they are achieved by using the proposed intrusion detection model.

Flexibility: Agents allow the learning capabilities to be incorporated in a natural way for the node and network resources. This helps in changing the policies of the mobile agents such as collection and distribution of intrusion information. HIDA can be embedded with neural network or other machine learning techniques.

Scalability: The proposed scheme is scalable to any size of a network, since agents detect intrusions in a distributed manner and are limited in travelling due to the limitations on the number of hops it can travel.

Efficiency: The scheme increases the efficiency of the network by detecting intrusions based on the activity patterns of the user, thereby reducing the entire intrusion information distribution across the network by using distributing mobile agents.

Adaptability: Functioning of the scheme itself justifies this feature. The node adapts to changes in the network conditions such as excluding masquerading nodes by distributing intrusion information across the network. The scheme assists in performing routing using only reliable and secured nodes by avoiding

routes through masqueraders.

Reusability: Reusability of the software can be seen in two categories: part of the mobile agent software and the algorithm software. For example, distributing mobile agent can be reused for discovering resources within a network and create a local map of the resources.

Maintainability: We can easily debug the agent components and also replace the old agents' components with new ones without affecting the other components.

Encapsulation: A mobile agent can be coded to perform aggregate tasks such as bandwidth measurement and allocation, delay estimation, loss detection, negotiating service level agreements, network behavior predictions, etc.

6. Conclusions

Wireless ad hoc networks have brought about a paradigm shift in the way we think about intrusion detection. We need to rethink intrusion detection methods for these networks based on their characteristics. In this paper, we proposed a distributed and agent based architecture for MANET IDS. Agent technology offers much to the field of intrusion detection. There are three main reasons for using agents to perform intrusion detection: performance enhancements, IDS design improvements and Response improvements. And also the main objective of the agent-based approach is to exploit the asynchronous interaction and Component Based Software Engineering (CBSE) feature of the agent technology for intrusion detection.

The proposed work demonstrates that activity pattern learning can be a valuable technique in the domain of anomaly detection for user recognition in computer security. The performance of the algorithm can be improved to detect different types of exploits like privilege escalation, removable media, export via email, changing file extensions encipher/decipher, and unusual search by considering semantics and the arguments used for an activity.

Acknowledgement

We thank the anonymous reviewers for giving us good suggestions to improve the quality of the paper.

References

[1] Jun-Zhao Sun, Mobile Ad Hoc Networking: An Essential Technology for Pervasive Computing, www.mediateam.oulu.fi/publications/pdf/92.pdf.

[2] Li and Wie, Guide Lines on Selecting Intrusion Detection Methods in MANET, isedj.org/isecon/2004/3233/isecon.2004.Li.pdf.

[3] Theuns Verwoerd and Ray Hunt, Intrusion Detection Techniques and Approaches, *Computer communications* **25**(15) (2002), 1356–1365.

[4] Amitabh Mishra, Ketan Nadkarni and Animesh Patcha, Vrginia Tech, Intrusion Detection in Wireless Ad hoc Networks, *IEEE Wireless Communications* (2004), 48–60.

[5] M. Crosbie and G. Spafford, Active Defence of a Computer system using Autonomous agents, Technical report 95-008, COAST Group, Department of Computer Sciences, Purdue University, West Lafayette, IN, 1995, 47907-1398.

[6] Y. Zhang and W. Lee, Intrusion detection in wireless Ad Hoc Networks, Proceedings of 6th Int'l Conf. Mobile Comp. And Net., 2000, 275–283.

[7] S. Bhargava and D.P. Agrawal, Security Enhancements in AODV protocol for Wireless Ad Hoc Networks, *Proc. VTC 2001 Fall* **4** (2001), 2143–2147.

[8] R. Ramanujan et al., Techniques for Intrusion-Resistant Ad Hoc Routing Algorithms (TIARA), *Proc. MILCOM* **2** (2000), 660–664.
[9] P. Albers et al., Security in Ad hoc Networks: a General Intrusion Detection Architecture Enhancing Trust Based Approaches, Proc. 1st Int'l Wksp. Wireless Info. Sys., Ciudad Real, Spain, 2002.
[10] O. Kachirski and R. Guha, Intrusion Detection Using Mobile Agents in Wireless Ad Hoc Networks, *Knowledge Media Net.*, (Piscataway, NJ, USA), IEEE Press, July 2002, 153–158.
[11] A.B. Smith, An Examination of an Intrusion Detection Architecture for Wireless Ad Hoc Networks, *Proc. 5th Nat'l. Colloq. for Info. Sys. Sec. Education*, 2001.
[12] Roy A. Maxion and Tahlia N. Town send, Masquerade Detection Using Truncated Commands Lines, *International Conference on Dependable Systems and Networks*, Washington, DC, 2002, 23–26.
[13] W. DuMouchel, Computer intrusion detection based on Bayes factors for comparing command transition probabilities, Technical Report 91, National Institute of Statistical Sciences, Research Triangle Park, North Carolina, 27709-4006, 1999.
[14] M. Schonlau and M. Theus, Detecting Masqueraders in intrusion detection based on unpopular commands, *Information Processing letters* **76**(1) (2000), 33–38.
[15] T. Lane and E.E. Broadley, Temporal sequence Learning and data reduction for Anomaly Detection, *ACM Transactions on Information and system Security* **2**(3) (1999), 295–331.
[16] M. Schonlau, W. DuMouchel, W.H. Ju, A.F. Karr, M. Theus and Y. Vardi, Computer Intrusion: Detecting Masqueraders, *Statistical Sciences* **116**(1) (2001), 58–74.
[17] S.S. Manvi and P. Venkataram, Applications of agent technology in communications: A Review, *Computer communications* **27**(15) (2004), 1493–1508.
[18] T. Karygiannis, Network Security Testing Using Mobile Agents, *Proceedings of the International Conference on Telecommunications*, Chalkidiki, Greece, 1998, Csrc.nist.gov/mobilesecurity/Publications/Agents_PAAM98.pdf.
[19] Yi-an Huang and Wenkee Le, A co-operative Intrusion Detection System for Ad Hoc Network, *Proc of the 1st ACM work shop on Security of ad hoc and sensor networks*, Fairfax, Virgna, 2003, 135–149.
[20] C.R. Lin and M. Gerla, Adaptive Clustering for Mobile Wireless Networks, *IEEE Journal on Selected Areas in communications* **15**(7) (1997), 1265–1275.
[21] B.M. Reshmi and S.S. Manvi, Masquerade detection by using activity patterns, Proc. European Conference on Computer Network Defence, Pontypridd, U.K., Dec. 2005, 241–250.
[22] K.P. Menelaos, G.C. Fotis, S.V. Iakovos et al., Mobile agent standards and available platforms, *Computer Networks* **31** (1999), 1999–2016.
[23] Laurentiu Lucian Petrea and Dan Grigoras, Mobile code platform for ad hoc networks, *Proc. International Symposium on Parallel and Distributed Computing*, De lille, France, July 2005.
[24] G. Derbas, A. Kayssi, A. Chehab, H. Artail and A. Tajeddine, A trust model for distributed systems based on reputation, *International Journal of Web and Grid Services*, Inderscience Publishers, **1**(3/4) (2005), 416–447.
[25] Yan Wang and Vijay Varadharajan, Secure route structures for parallel mobile agents based systems using fast binary dispatch, *Mobile Information Systems*, IOS Press, **1**(3) (2005), 185–205.

B.M. Reshmi obtained her post graduate degree from Basaveshwara Engineering College, Bagalkot, India, in 2005. She is currently working as an Assistant Professor in Information Science and Engineering Department, Basaveshwara Engineering College, Bagalkot, India. Her areas of interest include computer communications, wireless networks, agent technology applications and computer security. She has remained active in her areas of interest by publishing several papers in referred conferences. She is a member of ISTE, India.

S.S. Manvi obtained his doctoral degree from Indian Institute of Science, Bangalore, India in 2004. He is currently working as a Professor and Head of Electronics and Communication Engineering Department, Basaveshwara Engineering College, Bagalkot, India. His areas of interest include computer communications, wireless networks, multimedia communications, grid computing, e-commerce, and computational intelligence in tele communications, agent technology applications, and computer security. He has remained active in his areas of interest by authoring several papers in refereed conferences and journals. He has also coauthored a book on "communication Protocol Engineering", Published by PHI, 2004. He is a member of IEEE, USA, Fellow of IETE, India, and Member of ISTE, India. He has been included in Marquis Who's Who in the World, USA, international Biographee of Cambridge, London, for his noteworthy research contributions in the area of communications and networking.

Bhagyavati obtained her doctoral degree from University of Louisiana at Lafayette in 2001. She is currently working as an Assistant Professor in Computer Science Department, Columbus University, USA. Her areas of interest include Information assurance, infrastructure security and grid computing. She has remained active in her areas of interest by obtaining grants and authoring several papers in refereed conferences and journals. She is a member of the IEEE, ACM, Sigma XI and Upsilon, Pi Epsilon organizations