Advances in Mathematical & Computational Sciences Journal, Vol. 6 No. 2, June, 2018

Article Citation Format

Nwaocha, V.O. (2018). Agent, Multi-agent Technologies and Intrusion Detection Architecture in Mobile Ad-Hoc Networks (MANET. Advances in Mathematical & Computational Sciences Vol. 6 No. 2. Pp 87-107
https://www.isteams.net/mathematics-computationaljournal

Article Progress Time Stamps

Article Type: Research Article
Manuscript Received: 11st Apr, 2018
Review Type: Blind
Final Acceptance:: 12th March, 2018
dx.doi.org/10.22624/AIMS/MATHS/V6N2P7

Agent, Multi-agent Technologies and Intrusion Detection Architecture in Mobile Ad-Hoc Networks (MANET

Nwaocha, V.O
Department of Computer Science
National Open University of Nigeria
Abuja, FCT, Nigeria
E-mail: onwaocha@noun.edu.ng

Software agents react with other entities in various settings across multiple platforms. They are applicable in various fields such as in teaching, learning, industry, simulation, virtual reality, network security and more recently, in the software design of intrusion detection systems (IDSs). A multi-agent intrusion detection system is a set of autonomous components which work together in cooperation to detect intrusions. In this paper, we x-rayed the attributes of agents and multi-agent technologies such as mandatory and orthogonal properties. These properties make the agents different from the standard software. Set properties of agents such as autonomy, reactive, proactive, and temporally continuous are also examined while also evaluating intrusion detection systems architecture in Mobile Ad-Hoc networks.

Keywords: Agent, Multi-agent Technologies, Intrusion Detection Architecture, MANET and Networks

1. INTRODUCTION

A software agent is an autonomous entity that can interact with its environment to achieve a specific goal. Typically, software agents react with other entities in various settings across multiple platforms. They are applicable in various fields such as in teaching, learning, industry, simulation, virtual reality, network security and more recently, in the software design of intrusion detection systems (IDSs). A multi-agent intrusion detection system is a set of autonomous components which work together in cooperation to detect intrusions. Agents have some special; properties such as mandatory and orthogonal properties [14]. These properties make the agents different from the standard software. Set properties of agent are autonomy, reactive, proactive, and temporally continuous. Meanwhile, features of agents within the context of an ad hoc network setting are as follows:

i. 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 neighbours.

Discover routes in anticipation to link breaks, checking intruders, studding legitimate user behavior patterns etc. This allows them to operate independently.

ii. 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 neighbours of a node, etc.

iii. Pro-active and goal oriented: Agents anticipate the changes in the MANET environment and take appropriate decision. A key factor with agents is that they typically have a sinle task to complete, such as monitoring user behavior, monitoring for a user login, etc. This gives agent a small foot-print and also make them easier to test, and are more robust.

Generally, mobile agents differ from static agents in terms of mobile property. Hence, a mobile agent 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 [41]. 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 reliable. Normally, mobile agents can interact with environment by communicating with other static and mobile agents in the network.

Mobile agents imply agents in the network.

i. Reduce the network load.
ii. Reduce latency.
iii. Embedded protocols
iv. They interact with their environment and adapt themselves.
v. Move autonomously.

Meanwhile, multi-agent systems consist of multiple agents that interact and work together to achieve a particular set of tasks or goals [43]. The pervasive nature of such systems as a single agent acting on its own are assessed and addressed to achieve a desired objective through regular iteraction between distributed agents. Such an objective may include the protection of an international border against trespassing, timely detection of a bushfire, accurate analysis of traffic state of a large metropolis, remote monitoring of a vehicle conveying critical goods from source to destination and so forth.

The outcome of such collaboration between multiple agents is the aggregate property of a system as a whole and not a single agent [44]. Currently, agent and multi-agent system technologies, methods, and theories are being employed in diverse domains [45]. These include information retrieval, user interface design, robotics, electronic commerce, computer mediated collaboration, network security, computer games, education and training, smart environment, ubiquitous computers, and social simulation. They are not only very promising technologies, but also emerging as new way of thinking, a conceptual paradigm for analyzing problems, designing systems, and dealing with complexity, distribution and interactivity, and perhaps a new perspective on computing and intelligence.

1.1 Intrusion Detection System

The term intrusion simply refers to any set of action that attempt to compromise the confidentiality, availability or integrity of a resource [46]. Furthermore, intrusion detection can be described as a process of monitoring activities in a system which can be a computer or a network. Normally, intrusion detection works on the basis of examining the activity on a host or network and determining if that activity is normal known as the intrusion detection system (IDS). Generally, intrusion detection systems were introduced in order to detect possible violations of a security policy by monitoring system activities and response. For this reason, intrusion detection systems are aptly referred to as the second line of defense.

1.2 Classification of Intrusion Detection Systems

Intrusion detection systems are classified based on a number of criteria. Two decisive factors that determine the taxonomy of intrusion detection systems are as follows:

i. Host-based intrusion detection system (IDS)
ii. Network-based intrusion detection system (IDS)

Host-based intrusion detection systems use operating system or application logs in its analysis. They directly monitor the computer on which they run often through tight integration with the operating system. Audit data from a single host in used to detect intrusions. They monitor insiders with the same vigilance as outsiders, and network encryption doesn't affect them. But the number and diversity of computers often make it impossible to protect each computer individually with a host-based ID system. On the other hand network-based intrusion detection systems capture and analyze packets from network traffic between hosts. In this approach, network traffic data, along with audit data from one: or more hosts, is used to detect intrusions. Unlike host-based ID systems, which are out rightly, detecting malicious dehaviour, these systems deduce behavior based on the content and format of data packets on the network. Among other things, they analyze over requests for sensitive information and repeat failed attempts that violate security policy.

Generally, many of the existing host-base and network-based intrusion detection system perform data collection and analysis centrally using a monolithic architecture. In other words data is collected by a single host, either from audit trails or by monitoring packets in a network and analyzed by a single module using different techniques.

Hence some researchers have identified a number of issues [45] associated with theses architectures as follows:

i. The central analyzer is a single point of failure:
ii. Scalalability is limited
iii. It is difficult to reconfigure or add capabilities to the IDS;
iv. Analysis of network data can be flawed.

1.3 Intrusion Detection Systems on Detection Techniques

On the basis of detection techniques, IDS can also be classified into three categories as follows [46]:

i. Anomaly- based intrusion detection systems
ii. Misuse intrusion detection systems
iii. Specification-based intrusion detection system

(i) Anomaly detection system: The normal profiles of user are kept in the system. The system compares the capture data with these profiles, and them treats any activity that deviates from the baseline as a possible intrusion by informing system administrators or initializing a proper response. Anomaly intrusion detection systems have been shown to be effective for unknown or novel attacks since no prior knowledge about specific intrusions is required. Nevertheless, the main drawback of this approach is that they tend to generate more false alarm than do misuse detection. Another disadvantage of anomaly detection for mobile computing is that the normal profile must be periodically updated and the deviations from the normal profile computed. The periodic calculations can impose a heavy load some resource comparatively les computation might be better suited.

(ii) Misuse intrusion detection system: These systems keep patterns (or signatures) of known attacks and use them to compare with the captured data. Any matched pattern is treated as an intrusion. However, this sort of system does not detect new kinds of attacks.

(iii) Specification-based detection: The system defines a set of constraints that describe the correct operation of a program or protocol. Then, it monitors the execution of the correct program with respect to the defined constraints.

1.4. IDS Terminologies

Common security terms which are related to intrusion detection techniques are described as follows [47]:

- Vulnerability: Vulnerability is described as a weakness that allows an attacker to reduce the security of a particular system in a network. It is also considered as an "attack surface".
- Ii Exploit: An exploit is a piece of software or mechanism which takes advantages of bug or vulnerabilities that exits in the system in order to cause inadvertent behavior of the system. For instance, If poor passwords are used in network for authentication then a password-cracking might be the exploit on such vulnerability.
- Signature: Signatures are pattern sets which are used by IDS to identify an unwanted packet. A signature is usually created to watch network traffic for a particular attack or vulnerability.
- Iv Alarm: An alarm is considered as a signal generated by IDS in response of occurrence of an attack.
- Detection rate: The detection rate refers the fraction of all attacks that are actually detected.
- False alarm: A false alarm is an attack alarm that is triggered incorrectly. In other words, traffic that does not constitute an actual attack.
- False alarm: The false alarms rate (fraction of all normal data that produces (false) alerts.
- False Negative: A false negative is a term which means no alarm is triggered if any attack occurs. This is one of the worst types of false alarms.
- False Negative Rate: This is the quantity of illegitimate traffic wrongly detected as malicious.
- True Alarms: There are two types of true alarms triggered in IDS i.e. true positive and true negative.
- True Positive: A true positive is a type of alarm that is triggered when the IDS device has recognized and responded to an attack.
- True Negative: A true negative implies that an attack had occurred but IDS had not triggered an alarm.
- Packet delivery ration: This is the ratio of the total number of packets delivered to the total number of packets received in the system.

2. INTRUSION DETECTION IN MANETS

Although various researchers have provided a huge spectrum of research works on intrusion detection systems (IDSs) for wired networks [48], yet their implementation in MANETs make such IDSs ineffective and inefficient due to the specific features of MANETs. Consequently, researchers have sought more effective intrusion detection systems for mobile ad hoc networks. Besides, MANETs require added security mechanisms since attackers easily infiltrate this network through the subverted nodes. The vital issues that make applying existing solutions impractical are: Dynamic nature of MANETs, the absence of fixed infrastructure and resource constrained nodes. Thus, the aforementioned issues should be addressed while designing IDS for MANETs However, adaptation of existing solutions to MANETs, is the challenge to modem researchers.

2.1 Classification of IDS for MANETs

Intrusion detection systems for mobile ad hoc networks can be broadly classified into two main disciplines: design of the architecture of IDS and detection mechanisms. This section provides an overview of the common intrusion detection system architectures and related various detection schemes for MANETs. For each class of intrusion detection system, the architecture and the related functions are presented and analyzed focusing on their operational strengths and weaknesses.

2.2 Intrusion Detection Architectures for MANETs

Researchers have proposed several intrusion detection system architectures. According to [49], existing intrusion detection (IDS) architectures for MANETs fall under three main categories:

i. Stand-alone Intrusion Detection System (IDS) Architectures;
ii. Cooperative Intrusion Detection System (IDS) Architectures;
iii. Hierarchical Intrusion Detection System (IDS) Architectures;

3. STAND-ALONE INTRUSION DETECTION SYSTEM ARCHITECTURES

The stand-alone IDS architecture aims at detection malicious activities in a MANET using a self-contained approach [50]. This sort of architecture employs an intrusion detection engine installed at each node employing only the node's local audit data. Several authors have evaluated the most recent stand-alone IDS architectures for MANET, taking into consideration the strength and weaknesses of each one. Jacoby add Davis proposed a stand-alone architecture for detecting malicious actions in MANETs, by monitoring power consumption in every nod's battery [51]. Detection is achieved by comparing a node's power consumption with a set of power consumption patterns induced by known attacks, using smart battery technology.

In an experimental implementation, the proposed IDs detected 99% of the attacks in cases that only one sort of this attack occurred. It also detected multiple attacks, but only in cases that the nodes were idle and no other activity was present, The main advantage of this architecture is that it is more reliable (i.e., since it is based on hardware operation), compared to other IDSs that rely on audit data and anomaly-based detection, as these can be more easily manipulated by intruders. On the other hand, it detects only attacks that cause power consumption irregularities and only in cases that the nodes are idle, something that rarely occurs in real systems.

Nadkarni and Mishra [52] proposed a stand-alone IDS architecture that uses compound detection aiming at reducing the amount of false alarm alerts, which typically appear in anomaly detection. It employs adjusting thresholds to determine malicious behaviors. During initialization, the intrusion detection engine installed in every node creates the normalcy profile of the network traffic. Based on this, it estimates threshold values and beyond which there is an indication of possible attacks.

Every time a symptom of a known attack is detected, a counter called mis incident is incremented and the node responsible for the symptom is marked as suspicious. If the incident repeats and the mis-incident counter exceeds the threshold value for the specific attack, the node from where the incident originates is labeled as malicious. After a pre-set period of time in which there are no malicious behaviors detected, the threshold is raised; otherwise is lowered.

The most important strength of this architecture is that it is adaptable to network changes, because of the use of variable thresholds. Typically, periodic symptoms of suspicious behaviors, caused by network topology changes, will remain under the detection thresholds; while malicious behaviors that are constant will exceed the thresholds indicating the occurrence of attacks. On the other hand, the use of adjusting thresholds introduces new security weaknesses, since malicious node may exploit this mechanism. More specifically, a malicious node may increase the threshold values by performing legitimately for a certain period of time. Then, if the threshold values are high enough, it may perform an attack considering not exceeding the threshold values and raising alarms. Nodes that might not cooperated in the routing process or generate invalid routing updates due to out-dated routing information (i.e., caused by high mobility) might be falsely characterized as malicious, Moreover, coordinated attacks (i.e., such as byzantine attacks) cannot be detected, since nodes do not cooperate.

Finally, Adrian and his team [53] proposed a two-stage, stand-alone IDS architecture that aims at operating in resource-constrained environments, such as MANETs. It installs two different detection engines in every node, where the first one commonly referred to as the maximima detection system (MDS) is used to rapidly identify a potential threat and calibrate the second engine known as the cross-correlative detection system (CCDS). MDS is an anomaly detection engine that identifies statistical oddities in the observed interaction of the application layer. This is achieved by maintaining the history of the application layer interactions and comparing them with a normalcy profile created offline. If a possible attack is identified, MDS activates CCDS that calibrates a threshold value considering the attack.

Subsequently, the average values of the application behavior of every node are calculated and compared with the threshold. Behaviors that that exceed the threshold are marked as malicious. By employing two detection engines at each node, the proposed IDS increases detection accuracy, compared to other single engine IDSs because the one engines supplements the other. However, CCDS is prone to false alarms and negatives, since it calibrates the threshold value only once during start-up. Hence, dynamic changes of the network, induced by nodes mobility, are note accommodated by CCDS. The stand-alone IDS architectures for MANETs are summarized in Table 2.4 for easy comparison.

Table 1: Strength and weaknesses of the Stand-alone IDS architectures

IDS architecture	Strengths	Weaknesses
Battery-based IDS	Reliability, since it is based on hardware operations	It detects only attacks that cause power irregularities
Threshold-based IDS	Adaptability to network changes using adjusting thresholds	Introduces new security weaknesses It is prone to false alarms Cannot detect coordinated attacks
Two-stage IDS	Increased detection accuracy by employing two detection engines at each node.	It is prone to false alarms and negatives

4. COOPERATIVE IDS ARCHITECTURES

In the cooperative IDS architectures, and intrusion detection engine is installed of each node with a provision of local audit data monitoring and intrusion detection [54]. Cooperative architectures include an intrusion detection engine installed in every node, which supervise local audit data and exchanges audit data and or detection result with neighboring nodes in order to resolve inconclusive detections.

In the cooperation IDS architectures an intrusion detection is installed in every node monitoring local audit data and providing intrusion detection. To resolve inconclusive intrusion detections and detect more accurately advanced types of attacks, detection engines may cooperate with engines of neighboring nodes through the exchange of audit data or detection outcomes.

i. Cooperative IDS architecture based on social network analysis
ii. A multi-layer cooperative detection architecture
iii. Fork: A two proged intrusion sceme for MANETs
iv. Routing anomaly detection architecture
v. Layered intrusion detection framework for ad-hoc networks (LIDF)

4.1 Cooperation IDS architecture based on social network analysis

A team of researchers proposed a cooperative IDS architecture, which relies on a detection engine that utilizes social network analysis methods. In this architecture, each node deploys an intrusion detection engine that performs detections using audit data received from its "ego" network. An "ego" network consists of a hosting node ("ego") and the node ("alters") that are directly connected to it. The deployed engines operate similarly to anomaly detection, but they utilize social relations as metrics of interest, which require less computational overhead compered to standard anomaly detection engines [55]. Moreover. A training phase is also required to create normal profiles (i.e., in anomaly detection), and according to the authors, the detection engines monitor the Medium Access Control (MAC) and network layers.

The propose IDS is compose of three modules: (a) the data pre-processing module that collects and pre-processes audit; (b) the social analysis module that performs intrusion detection; and (c) the response module that integrates local and global (i.e., gathered from neighboring node) intrusion alert.

During the IDS operation, the data pre-processing module collects audit data from its neighboring nodes in intervals of five seconds. The social analysis module, then, processes the collected data in order to realize social analysis module, then processes the collected represent the behavior of these nodes at a certain time.

Subsequently, the relations accomplished are compared to the normal profile of expected behaviors, and any variation from these constitutes an intrusion. If an intrusion is detected, the response module notifies the neighboring nodes. The main strength of this architecture is that the employed detection engines incur less computation complexity; compared to conventional anomaly detection engines [55]. On the other hand, it presents some weaknesses outlined as follows:

i. The rate of false alarms may increase in the detection accuracy may drop in cases of high period of time to create social relations with neighboring nodes, before it changes its location. As a result, there would not be enough information for the social analysis module to distinguish between normal and malicious behaviors.
ii. Audit data exchange may increase the communication load among nodes, causing degradation to the network performance. The authors have arbitrarily selected a five second interval for audit data exchange within "ego" network, without any evaluation of the impact of this parameter to the network performance.
iii. New security risks may arise from the exchange of audit data, since a malicious node may either transmit false audit data or avoid transmitting any of them, in order to hinder or mislead the detection process.

4.2 A multi-layer cooperation detection architecture

A group of scholars [56] proposed a cooperative IDS architecture that uses that uses three parallel anomaly detection engines, referred as MAC layer detection engine, routing detection engine, and application layer detection engine, installed in every node. The use of multi-layer detection aims at increasing detection accuracy, since attacks that target upper-layer protocols can be seen as legitimate events at lower-layers, and vice versa. The MAC layer detection engine monitors both access control and addressing at the data link layer. The routing detection engine monitors the network layer and keeps track of the packet delivery and outing information. Finally, the application layer engine monitors the application layer. Each engine collects the appropriate audit data, process them and looks for malicious behaviors within them. In every node, while a global integration module combines the results received from the neighboring nodes. A set of simulations has been performed using GloMoSim to evaluate the effectiveness of the proposed architecture.

The multi-layer IDS presents the following strengths:

i. It increase the detection accuracy, compared to other single engine detection solutions, as the multiple detection engines supplement each other. I the simulation results, the detection accuracy increased up to 20% through integrating the results of all three engines, compared to the results that each detection engine yielded by itself [56].
ii. Although it uses cooperation between the neighboring nodes, it induce relatively low communication overhead, since only the detection results and not the voluminous audit data are exchange.

The cooperative IDS architectures considered in the section presents some weaknesses:

i. Its operation increase the processing overhead in each node, compared to other single engine solution, since the IDS deploys three detection engines instead of one. So far, the authors have not studied or evaluated the processing overhead of the proposed architecture.
ii. The ratio of false alarms and the detection accuracy of the IDS are negatively affected by high packet loss and/ or high nodes' mobility. This is because the routing detection engine relies on packet delivery and routing information to detect attacks. Except for the local integration modules, the inaccurate detection results also influence the global integration modules of the neighboring nodes.
iii. The functionality of cooperation creates new security risks, since a malicious node may either transmit false detection results (i.e., "blackmail" attack) or modify detection results originating from another cooperating node (i.e., "man in the middle attack") in order to hinder or mislead the detection process in a node or set of nodes.

4.3 Fork: A two pronged intrusion detection scheme for MANETs

A team of researchers, have proposed a cooperative IDS architecture [57], which uses lightweight modules (agents) able to perform different detection tasks and aim at reducing battery consumption. Each network node contains all the modules required to perform the detection tasks and is assigned a reputation value, which increases when the node successfully assists with intrusion detection is unsatisfactory. Nevertheless, the authors do to clarify under what conditions the node's performance is deemed unsatisfactory. The employed intrusion detection engine relies on anomaly detection and it is installed in every node. When the engine of a node detects a suspicious behavior, it initiates an auction scheme to select a set of nodes that are most suitable to assist in performing intrusion detection. Nodes with the highest amount of battery resources and reputation value are selected and specific task are assigned to them.

These tasks include:

i. The execution of host or network monitoring
ii. The decision making given a set of audit data and
iii. The activation of defensive actions in case that malicious behaviors have been detected. The authors neither collaborate on how nodes' cooperation is achieved nor evaluate the communication overhead imposed by the employed cooperation mechanism. Moreover, they did not consider node's mobility in the performed simulations, thus the impact of mobility on the detection accuracy, the rate of false alarms and the communication overhead cannot be determined. The main advantage of the Fork architecture is the distribution of detection tasks among a set of nodes, which reduces the processing load for the initiating node and conserves its battery power. The selection of assisting nodes also considers, among other criteria, the available battery resources thus, node with lower battery power are not burdened with intrusion detection responsibilities.

On the other hand, the weaknesses of the architecture are enumerate as follows:

i. High nodes' mobility typical increases the communication overhead impose by the IDS architecture. A node assigned with a detection task may move away from the initiating node thus, it has to route the results regarding its tsk through other nodes.
However, This extra communication overhead has not been quantified through a simulation or analytic study.

ii. It is vulnerable to man in the middle attacks, since a malicious node exploiting the task allocation mechanism, may capture and modify intrusion detection task messages. A malicious node might also cause blackmail attacks, by transmitting false detection results to the node that has initiated detection tasks. Finally, a malicious node may cause sleep deprivation attacks, by initiating fake tasks to other nodes in order to consume their resources.

4.4 Routing anomaly detection architecture

Sun et al. [have proposed a cooperative IDS architecture that focuses on routing disruption attacks. Since all the nodes of a MANET participate in routing, each one maintains a table that contains routing information, such as routing paths to reach other nodes and the required number of hops. Extensive changes in this table may be a symptom of malicious behaviors that attempt to disrupt the routing process. The proposed IDS uses the following two routing features to discover malicious behaviors:

i. The percentage of in the route entries (PCR), and
ii. The percentage of change in the number of hope (PCH). PCR represents the added/deleted route entries during a certain period of time , while PCH indicate the change in the sum of hope of all route entries over the period of time

In this intrusion detection system, one or several intrusion detection engines that rely on anomaly detection are installed in every node. These engines collect and process routing information to detect possible intrusions, using a modified Markov Chain anomaly detection method [59]. In the event that several detection engines are deployed in a node. Alerts and reports from each local engine are combined. Moreover, data reports and alerts from neighboring nodes are also correlated in order to attain more accurate decisions. Based on the performed simulations, the authors state that this IDS detects more than 90% of the routing disruption attacks, in scenarios with relative low nodes' mobility. The main advantage of this architecture is related to the increased detection accuracy that it presents, because of the deployment of multiple detection engine at each node (i.e. compared to other single engine solutions). This fact also makes this IDS fault tolerant in cases that a detection engine fails or becomes a target of an attack.

On the other hand, it presents some drawbacks:

i. It cannot be used to detect all the types of possible attacks, since it monitors only for routing attacks
ii. It imposes extra communication overhead, since detection engines hosted at neighboring nodes have to costly exchange detection reports and alert in order to reach more accurate decisions.
iii. The detection accuracy and the ratio of false alarms are negatively affected by nodes' mobility. This occurs for two reasons:
    a. In a high mobility scenario, a node would only notice a few falsified routing changes before changing its location; and
    b. In such scenarios, the changes in routing tables are rapid and inconsistent. Thus, there is not enough information for the detector to distinguish between normal behaviors provoked by nodes mobility and abnormal behaviors provoked by malicious nodes
iv. It vulnerable to blackmail attacks, since a malicious mode might transmit false detection repors or alerts in order to hinder the intrusion detection process and falsely accuse a legitimate node(s) as malicious. Later on. Sun set al. [21] improved the aforementioned routinig anomaly IDS architecture, by adjustable thresholds. This addresses some of the most important drawbacks of this architecture, such as the negative impacts of nodes' mobility on the detection accuracy and the ratio of false alarms.

4.5 Layered intrusion detection framework for ad-hoc networks (LIDF)

Komninos and Douligeris have propose a cooperative IDS architecture [60]. Which relies on multilayered detection to capture malicious behaviors. In this architecture. Every host maintains an intrusion detection unit, which is divided into three modules:

a. The collection
b. The detection and
c. The alert module.

The collection module is responsible for collecting audit data from both the data link and the network layer. By monitoring these two layers the IDS has a close view of the networking activities. The detection module performs anomaly- based detection on the collected audit data in two steps, in order to conserve the host's resources and battery. First, it processes only the most recent local audit data. In case that these data are not sufficient to reach an accurate decision regarding a suspicious behavior, more audit data are requested from neighbouring nodes via secure communication channels. However, the authors have not specified when nodes make a decision of requesting nieghbours' cooperation overhead imposed by nodes' cooperation cannot be determined. Ultimately, in case that a malicious behaviours is detected, the alert module has the responsibibility to notify the neighbourring nodes.

The key strength of this sort of IDS architecture are as follows:

i. Using multiple layers of detection, it is able to detect attacks at both the network and data link layer
ii. The use of secure communication channels for nodes' cooperation defeats man in the middle attacks

On the other hand, the weaknesses of this architecture are:

i. It focuses only on attacks that target the network and data link layer. Attack at the transport layer-suchh as a SYN flooding, where a malicious node sends a large number of SYN packets, or a session hijacking attack, where a malicious node takes control over a session between two nodes-will go undetected.
ii. Nodes' mobility reduces the detection accuracy of the IDS and increase the ratio of false alarms, since it hinders cooperation as the nodes move away from each other.
iii. It is vulnerable to blackmail attacks, since a malicious node that cooperates might transmit modified audit data in order to hinder the intrusion detection process, hide malicious activities or falsely accuse legitimate nodes as malicious.

The vital strengths and weaknesses of the cooperative IDS architecture are summarized in Table 2.

Table 2: Strengths and weaknesses of the Cooperative MS architectures

IDS architecture	Strengths	Weakness
Cooperative IDS Architecture based On social network Analysis	The employ social based detection engine incurs less computational complexity than the conventional anomaly-based engines.	The ratio of false alarms and detection accuracy are negatively affected by high nodes' mobility. Audit data exchange increase the communication load among nodes Audit data exchange creates new security risks
Multi-layer Cooperative IDS	The multiple detection engines employed provide increased	The employment of multiple engines at each node increase the processing overhead.
Architecture	Detection	The ratio of false alarms and detection accuracy are negatively affected by high packet loss and 0or high nodes mobility
	The exchange of detection result among the neighboring nodes achieves nodes cooperation with the minimum communication overhead.	It is vulnerable to blackmail and man in the middle attacks
FORK	It reduces the processing load and conserves the battery power of nodes through task distribution	Then communication overhead is increased under high nodes mobility It is vulnerable to blackmail, man in the middle, and sleep deprivation attacks
Routing anomaly detection architecture	The multiple detection engines employed provide increased detection accuracy and a fault tolerant solution	In the initially propose architecture, the ratio of false alarms and detection accuracy are negatively affected by high node mobility It detect only routing attacks

		It impose extra communication overhead It is vulnerable to blackmail attacks
LIDF	It is able to detect attacks at multiple layers (i.e. network and data link layers) It defeats man-in- middle attacks using secure communication channels	It does not detect attacks at the transport layer (i.e. SYN flooding session hijacking etc). The ratio of false alarms and detection accuracy are negatively affected any high nodes' mobility It is vulnerable to blackmail attacks.

The following inferences can be drawn based on the strengths of the cooperative IDS architecture:

i. A good number of IDS with cooperative architectures employ multiple detection engines in order to provide increased detection accuracy and detect a wide set of possible attacks:
ii. Some of the attempt to minimize the imposed processing and communication overheads through task distribution or the exchange of detection results, instead of voluminous audit data among neighbouring nodes; and
iii. A few of them attempt to defeat certain attacks by employing trust or secure communication channels.

On the other hand, the following conclusions can be made on the basis of their weaknesses:

i. In the entire set of the studied architectures the ratio of false alarms and detection accuracy are negatively affected by nodes mobility
ii. Almost all of them impose extra processing and communication overhead and
iii. Most of them are highly vulnerable to network attacks as man in the middle, blackmail etc.

5. HIERARCHICAL IDS ARCHITECTURE

The hierarchical architedtures amount to a multilayer approach, which divide the network into clusters. Special nodes are selected to act as cluster-heads and undertake various responsibilities and roles in intrusion detection that are usually different from those of the simple cluster members. Similarly, in a MANET using a hierarchical IDS architecture, the nodes are divided into two categories: cluster-heads and cluster members. The cluster members run a lightweight local intrusion detection engine, while the cluster-head runs a comprehensive detection engine that processes pre-processed audit data from all the cluster members. This section describes some prominent hierarchical IDS architecture.

5.1 A Cluster-Based Intrusion Detection Architecture With Adaptive Selection Even Triggering

The hierarchical IDS architecture, proposed by ma and Fang [61]. Follows a modular approach based on clusters. The goal is to provide a clustered structure where cluster-heads are always hosted by nodes with the highest battery power. During network initialization, each node reports Its battery power to its neighbours. Then, the node with the highest available battery power is elected as cluster-head.

In the cooperation IDS architectures an intrusion detection is installed in every node monitoring local audit data and providing intrusion detection. To resolve inconclusive intrusion detections and detect more accurately advanced types of attacks, detection engines may cooperate with engines of neighboring nodes through the exchange of audit data or detection outcomes.

vi. Cooperative IDS architecture based on social network analysis
vii. A multi-layer cooperative detection architecture
viii. Fork: A two proged intrusion sceme for MANETs
ix. Routing anomaly detection architecture
x. Layered intrusion detection framework for ad-hoc networks (LIDF)

5.2 Cooperation IDS architecture based on social network analysis

A team of researchers proposed a cooperative IDS architecture, which relies on a detection engine that utilizes social network analysis methods. In this architecture, each node deploys an intrusion detection engine that performs detections using audit data received from its "ego" network. An "ego" network consists of a hosting node ("ego") and the node ("alters") that are directly connected to it. The deployed engines operate similarly to anomaly detection, but they utilize social relations as metrics of interest, which require less computational overhead compered to standard anomaly detection engines [55]. Moreover. A training phase is also required to create normal profiles (i.e., in anomaly detection), and according to the authors, the detection engines monitor the Medium Access Control (MAC) and network layers.

The propose IDS is compose of three modules: (a) the data pre-processing module that collects and pre-processes audit; (b) the social analysis module that performs intrusion detection; and (c) the response module that integrates local and global (i.e., gathered from neighboring node) intrusion alert, During the IDS operation, the data pre-processing module collects audit data from its neighboring nodes in intervals of five seconds. The social analysis module, then, processes the collected data in order to realize social analysis module, then processes the collected represent the behavior of these nodes at a certain time.

Subsequently, the relations accomplished are compared to the normal profile of expected behaviors, and any variation from these constitutes an intrusion. If an intrusion is detected, the response module notifies the neighboring nodes. The main strength of this architecture is that the employed detection engines incur less computation complexity; compared to conventional anomaly detection engines [55]. On the other hand, it presents some weaknesses outlined as follows:

iv. The rate of false alarms may increase in the detection accuracy may drop in cases of high period of time to create social relations with neighboring nodes, before it changes its location. As a result, there would not be enough information for the social analysis module to distinguish between normal and malicious behaviors.
v. Audit data exchange may increase the communication load among nodes, causing degradation to the network performance. The authors have arbitrarily selected a five second interval for audit data exchange within "ego" network, without any evaluation of the impact of this parameter to the network performance.
vi. New security risks may arise from the exchange of audit data, since a malicious node may either transmit false audit data or avoid transmitting any of them, in order to hinder or mislead the detection process.

5.3 A Multi-Layer Cooperation Detection Architecture

A group of scholars [56] proposed a cooperative IDS architecture that uses that uses three parallel anomaly detection engines, referred as MAC layer detection engine, routing detection engine, and application layer detection engine, installed in every node. The use of multi-layer detection aims at increasing detection accuracy, since attacks that target upper-layer protocols can be seen as legitimate events at lower-layers, and vice versa. The MAC layer detection engine monitors both access control and addressing at the data link layer. The routing detection engine monitors the network layer and keeps track of the packet delivery and outing information. Finally, the application layer engine monitors the application layer. Each engine collects the appropriate audit data, process them and looks for malicious behaviors within them. In every node, while a global integration module combines the results received from the neighboring nodes. A set of simulations has been performed using GloMoSim to evaluate the effectiveness of the proposed architecture.

The multi-layer IDS presents the following strengths:

iii. It increase the detection accuracy, compared to other single engine detection solutions, as the multiple detection engines supplement each other. I the simulation results, the detection accuracy increased up to 20% through integrating the results of all three engines, compared to the results that each detection engine yielded by itself [56].

iv. Although it uses cooperation between the neighboring nodes, it induce relatively low communication overhead, since only the detection results and not the voluminous audit data are exchange.

The cooperative IDS architectures considered in the section presents some weaknesses:

iv. Its operation increase the processing overhead in each node, compared to other single engine solution, since the IDS deploys three detection engines instead of one. So far, the authors have not studied or evaluated the processing overhead of the proposed architecture.

v. The ratio of false alarms and the detection accuracy of the IDS are negatively affected by high packet loss and/ or high nodes' mobility. This is because the routing detection engine relies on packet delivery and routing information to detect attacks. Except for the local integration modules, the inaccurate detection results also influence the global integration modules of the neighboring nodes.

vi. The functionality of cooperation creates new security risks, since a malicious node may either transmit false detection results (i.e., "blackmail" attack) or modify detection results originating from another cooperating node (i.e., "man in the middle attack") in order to hinder or mislead the detection process in a node or set of nodes.

5.4 Fork: A Two Pronged Intrusion Detection Scheme For Manets

A team of researchers, have proposed a cooperative IDS architecture [57], which uses lightweight modules (agents) able to perform different detection tasks and aim at reducing battery consumption. Each network node contains all the modules required to perform the detection tasks and is assigned a reputation value, which increases when the node successfully assists with intrusion detection is unsatisfactory. Nevertheless, the authors do to clarify under what conditions the node's performance is deemed unsatisfactory. The employed intrusion detection engine relies on anomaly detection and it is installed in every node. When the engine of a node detects a suspicious behavior, it initiates an auction scheme to select a set of nodes that are most suitable to assist in performing intrusion detection. Nodes with the highest amount of battery resources and reputation value are selected and specific task are assigned to them.

These tasks include:

iv. The execution of host or network monitoring
v. The decision making given a set of audit data and
vi. The activation of defensive actions in case that malicious behaviors have been detected. The authors neither collaborate on how nodes' cooperation is achieved nor evaluate the communication overhead imposed by the employed cooperation mechanism. Moreover, they did not consider node's mobility in the performed simulations, thus the impact of mobility on the detection accuracy, the rate of false alarms and the communication overhead cannot be determined. The main advantage of the Fork architecture is the distribution of detection tasks among a set of nodes, which reduces the processing load for the initiating node and conserves its battery power. The selection of assisting nodes also considers, among other criteria, the available battery resources thus, node with lower battery power are not burdened with intrusion detection responsibilities.

On the other hand, the weaknesses of the architecture are enumerate as follows:

iii. High nodes' mobility typical increases the communication overhead impose by the IDS architecture. A node assigned with a detection task may move away from the initiating node thus, it has to route the results regarding its tsk through other nodes.
However, This extra communication overhead has not been quantified through a simulation or analytic study.
iv. It is vulnerable to man in the middle attacks, since a malicious node exploiting the task allocation mechanism, may capture and modify intrusion detection task messages. A malicious node might also cause blackmail attacks, by transmitting false detection results to the node that has initiated detection tasks. Finally, a malicious node may cause sleep deprivation attacks, by initiating fake tasks to other nodes in order to consume their resources.

6. ROUTING ANOMALY DETECTION ARCHITECTURE

Sun et al. [have proposed a cooperative IDS architecture that focuses on routing disruption attacks. Since all the nodes of a MANET participate in routing, each one maintains a table that contains routing information, such as routing paths to reach other nodes and the required number of hops. Extensive changes in this table may be a symptom of malicious behaviors that attempt to disrupt the routing process. The proposed IDS uses the following two routing features to discover malicious behaviors:

iii. The percentage of in the route entries (PCR), and
iv. The percentage of change in the number of hope (PCH). PCR represents the added/deleted route entries during a certain period of time , while PCH indicate the change in the sum of hope of all route entries over the period of time

In this intrusion detection system, one or several intrusion detection engines that rely on anomaly detection are installed in every node. These engines collect and process routing information to detect possible intrusions, using a modified Markov Chain anomaly detection method [59]. In the event that several detection engines are deployed in a node. Alerts and reports from each local engine are combined. Moreover, data reports and alerts from neighboring nodes are also correlated in order to attain more accurate decisions. Based on the performed simulations, the authors state that this IDS detects more than 90% of the routing disruption attacks, in scenarios with relative low nodes' mobility.

The main advantage of this architecture is related to the increased detection accuracy that it presents, because of the deployment of multiple detection engine at each node (i.e. compared to other single engine solutions). This fact also makes this IDS fault tolerant in cases that a detection engine fails or becomes a target of an attack. On the other hand, it presents some drawbacks:

v. It cannot be used to detect all the types of possible attacks, since it monitors only for routing attacks
vi. It imposes extra communication overhead, since detection engines hosted at neighboring nodes have to costly exchange detection reports and alert in order to reach more accurate decisions.
vii. The detection accuracy and the ratio of false alarms are negatively affected by nodes' mobility. This occurs for two reasons:
    c. In a high mobility scenario, a node would only notice a few falsified routing changes before changing its location; and
    d. In such scenarios, the changes in routing tables are rapid and inconsistent. Thus, there is not enough information for the detector to distinguish between normal behaviors provoked by nodes mobility and abnormal behaviors provoked by malicious nodes
viii. It vulnerable to blackmail attacks, since a malicious mode might transmit false detection repors or alerts in order to hinder the intrusion detection process and falsely accuse a legitimate node(s) as malicious. Later on. Sun set al. [21] improved the aforementioned routinig anomaly IDS architecture, by adjustable thresholds. This addresses some of the most important drawbacks of this architecture, such as the negative impacts of nodes' mobility on the detection accuracy and the ratio of false alarms.

6.1 Layered intrusion detection framework for ad-hoc networks (LIDF)

Komninos and Douligeris have propose a cooperative IDS architecture [60]. Which relies on multilayered detection to capture malicious behaviors. In this architecture. Every host maintains an intrusion detection unit, which is divided into three modules:

d. The collection
e. The detection and
f. The alert module.

The collection module is responsible for collecting audit data from both the data link and the network layer. By monitoring these two layers the IDS has a close view of the networking activities. The detection module performs anomaly- based detection on the collected audit data in two steps, in order to conserve the host's resources and battery. First, it processes only the most recent local audit data. In case that these data are not sufficient to reach an accurate decision regarding a suspicious behavior, more audit data are requested from neighbouring nodes via secure communication channels. However, the authors have not specified when nodes make a decision of requesting nieghbours' cooperation overhead imposed by nodes' cooperation cannot be determined. Ultimately, in case that a malicious behaviours is detected, the alert module has the responsibibility to notify the neighbourring nodes.

The key strength of this sort of IDS architecture are as follows:

iii. Using multiple layers of detection, it is able to detect attacks at both the network and data link layer
iv. The use of secure communication channels for nodes' cooperation defeats man in the middle attacks

On the other hand, the weaknesses of this architecture are:

iv. It focuses only on attacks that target the network and data link layer. Attack at the transport layer-suchh as a SYN flooding, where a malicious node sends a large number of SYN packets, or a session hijacking attack, where a malicious node takes control over a session between two nodes-will go undetected.

v. Nodes' mobility reduces the detection accuracy of the IDS and increase the ratio of false alarms, since it hinders cooperation as the nodes move away from each other.

vi. It is vulnerable to blackmail attacks, since a malicious node that cooperates might transmit modified audit data in order to hinder the intrusion detection process, hide malicious activities or falsely accuse legitimate nodes as malicious.

The vital strengths and weaknesses of the cooperative IDS architecture are summarized in Table 3.

Table 3: Strengths and weaknesses of the Cooperative MS architectures

IDS architecture	Strengths	Weakness
Cooperative IDS Architecture based On social network Analysis	The employ social based detection engine incurs less computational complexity than the conventional anomaly-based engines.	The ratio of false alarms and detection accuracy are negatively affected by high nodes' mobility. Audit data exchange increase the communication load among nodes Audit data exchange creates new security risks
Multi-layer Cooperative IDS	The multiple detection engines employed provide increased	The employment of multiple engines at each node increase the processing overhead.
Architecture	Detection	The ratio of false alarms and detection accuracy are negatively affected by high packet loss and 0or high nodes mobility
	The exchange of detection result among the neighboring nodes achieves nodes cooperation with the minimum communication overhead.	It is vulnerable to blackmail and man in the middle attacks
FORK	It reduces the processing load and conserves the battery power of nodes through task distribution	Then communication overhead is increased under high nodes mobility It is vulnerable to blackmail,

		man in the middle, and sleep deprivation attacks
Routing anomaly detection architecture	The multiple detection engines employed provide increased detection accuracy and a fault tolerant solution	In the initially propose architecture, the ratio of false alarms and detection accuracy are negatively affected by high node mobility It detect only routing attacks It impose extra communication overhead It is vulnerable to blackmail attacks
LIDF	It is able to detect attacks at multiple layers (i.e. network and data link layers) It defeats man-in- middle attacks using secure communication channels	It does not detect attacks at the transport layer (i.e. SYN flooding session hijacking etc). The ratio of false alarms and detection accuracy are negatively affected any high nodes' mobility It is vulnerable to blackmail attacks.

The following inferences can be drawn based on the strengths of the cooperative IDS architecture:

iv. A good number of IDS with cooperative architectures employ multiple detection engines in order to provide increased detection accuracy and detect a wide set of possible attacks:
v. Some of the attempt to minimize the imposed processing and communication overheads through task distribution or the exchange of detection results, instead of voluminous audit data among neighbouring nodes; and
vi. A few of them attempt to defeat certain attacks by employing trust or secure communication channels.

On the other hand, the following conclusions can be made on the basis of their weaknesses:

iv. In the entire set of the studied architectures the ratio of false alarms and detection accuracy are negatively affected by nodes mobility
v. Almost all of them impose extra processing and communication overhead and
vi. Most of them are highly vulnerable to network attacks as man in the middle, blackmail etc.

7. CONCLUDING REMARKS

The hierarchical architedtures amount to a multilayer approach, which divide the network into clusters. Special nodes are selected to act as cluster-heads and undertake various responsibilities and roles in intrusion detection that are usually different from those of the simple cluster members. Similarly, in a MANET using a hierarchical IDS architecture, the nodes are divided into two categories: cluster-heads and cluster members. The cluster members run a lightweight local intrusion detection engine, while the cluster-head runs a comprehensive detection engine that processes pre-processed audit data from all the cluster members. This section describes some prominent hierarchical IDS architecture. The hierarchical IDS architecture, proposed by ma and Fang [61]. Follows a modular approach based on clusters. The goal is to provide a clustered structure where cluster-heads are always hosted by nodes with the highest battery power. During network initialization, each node reports Its battery power to its neighbours. Then, the node with the highest available battery power is elected as cluster-head.

REFERENCES

1 [14] V.O. Nwaocha and H.C. Inyiama. "Securing Enterprise Networks: A Multi-agent Based Distributed Intrusion Detection Approach". International Journal of Computational Intelligence and Information Security, Vol. 4, No. 6. (2013) ISSN: 1837-7823

2 [41] M. Wooldridge.'An introduction to multi-agent systems". John Wiley and Sons; 2002.

3 [43] Weiss G. Multi-agent systems: a modern approach to distributed artificial intelligence.. The MIT Press; (1999).

4 [44] C. Krügel and T. Toth, "A Survey on Intrusion Detec-tion Systems," TU Vienna, Austria, 2000.

5 [45] A. K. Jones and R. S. Sielken, "Computer System Intru-sion Detection: A Survey," University of Virginia, 1999.

6 [46] K. Scarfone and P. Mell, "Guide to Intrusion Detection and Prevention Systems (IDPS)," NIST 800-94, Feb 2007.

7 [50] C.C. Xenakis, C. Panos and I. Stavrakakis, " A comparative evaluation of intrusion detection architectures for mobile ad hoc networks". Computer. Security, 30: 63-80.(2011).

8 [51] G.A. Jacoby, and N.J. Davis, " Mobile host-based intrusion detection and attack identification. IEEE Wireless Commun., 14: 53-60. (2007).

9 [52] K. Nadkarni, and A. Mishra. "A novel intrusion detection approach for wireless ad hoc networks. Proceedings of the IEEE Wireless Communications and Networking Conference, Volume 2, March 21-25, 2004, Atlanta, Georgia, USA., pp: 831-836.(2004)

10 [55] Bose, S. S. Bharathimurugan and A. Kannan, 2007. Multi-layer integrated anomaly intrusion detection system for mobile ad hoc networks. Proceedings of the International Conference on Signal Processing, Communications and Networking, February 22-24, Chennai, pp: 360-365. (2007).

11 [56] Razak, S.A., S.M. Furnell, N.L. Clarke and P.J. Brooke. "Friend-assisted intrusion detection and response mechanisms for mobile ad hoc networks". Ad Hoc Networks, 6: 1151-1167.2008).

12 [57] Ramachandran, C., S. Misra and M.S. Obaidat, 2008. FORK: A novel two-pronged strategy for an agent-based intrusion detection scheme in ad-hoc networks. Comput. Commun., 31: 3855-3869.

14 [59] Ma, C.X. and Z.M. Fang, 2008. A novel intrusion detection architecture based on adaptive selection event triggering for mobile ad-hoc networks. Proceedings of the IEEE 2nd International Symposium on Intelligent Information Technology and Security Informatics, January 23-25, 2008, Moscow, pp: 198-201.

13 [61] Marchang, N. and R. Datta, 2008. Collaborative techniques for intrusion detection in mobile ad-hoc networks. Ad Hoc Networks, 6: 508-523.