A NOVEL INTRUSION DETECTION SYSTEM FOR MANETS

Christoforos Panos[1], Christos Xenakis[2], Ioannis Stavrakakis[1]
[1]*Department of Informatics & Telecommunications, University of Athens, Panepistimioupolis, Ilisia, PC 15784, Athens, Greece*
cpanos@di.uoa.gr, ioannis@di.uoa.gr
[2]*Department of Digital Systems, University of Piraeus, Karaoli and Dimitriou 80, PC 18534, Piraeus, Greece*
xenakis@unipi.gr

Keywords: Intrusion detection system, IDS, mobile ad hoc networks, MANETs, random walks, specification-based intrusion detection.

Abstract: This paper proposes a novel Intrusion Detection System (IDS) for Mobile Ad Hoc Networks (MANETs) that aims at overcoming the limitations and weaknesses of the existing IDSs. The proposed IDS incorporates a novel random walk-based IDS architecture as well as a multi-layer, specification-based detection engine. The proposed solution does not belong to any of the existing intrusion detection approaches, since it relies on a set of robust, self-contained Random Walk Detectors (RWDs), which may freely move from node to node and randomly traverse a network, while monitoring each visiting node for malicious behaviour. RWDs exhibit a number of benefits including locality, simplicity, low overhead, and robustness to changes in topology. Moreover, the multi-layer, specification-based engine monitors the transport, network and data link layers of the protocol stack, providing an integrated solution capable of detecting the majority of security attacks occurring in MANETs.

1 INTRODUCTION

Mobile ad hoc networks (MANETs) are wireless networks, which operate without the aid of any established infrastructure or centralized authority. In MANETs, the nodes themselves implement the network management in a cooperative fashion and thus, all of them are responsible for this. MANET nodes have stringent resource constrains and they are typically mobile, forming a highly dynamic network topology, absent of any clear network boundaries. As a result, MANETs are susceptible to a variety of attacks such as eavesdropping, routing, packet modification, etc. (Djenouri et al., 2005), and securing a MANET under such conditions is challenging. An effective way to identify when an attack occurs in a MANET is the deployment of an Intrusion Detection System (IDS). An IDS is a sensoring mechanism that monitors nodes' and network activities in order to detect malicious actions and, ultimately, an intruder. An IDS can be divided into two parts: (i) the architecture that exemplifies its operational structure; and (ii) the detection engine that is the mechanism used to detect malicious behavior(s).

The existing IDS architectures for MANETs fall under three basic categories (Mishra et al., 2004): (a) stand-alone, (b) cooperative, and (c) hierarchical. The *stand-alone architectures* use an intrusion detection engine installed at each node utilizing only the node's local audit data. This fact (i.e., relying only on local audit data to resolve malicious behaviors) limits them in terms of detection accuracy and the type of attacks that they detect (Sen et al., 2009) (due to the distributed nature of MANETs) and, thus, we will exclude them from the analysis carried in section 2. On the other hand, the cooperative and hierarchical architectures process each host's audit data locally (i.e., similarly to stand-alone), but they also use *collaborative techniques* to detect more accurately a wider set of attacks. Thus, the majority of the most recent IDSs for MANETs are based on them (Sen et al., 2009). The *cooperative architectures* include an intrusion detection engine installed in every node, which monitors local audit data and exchanges audit data and/or detection outcomes with neighboring nodes, in order to resolve inconclusive (based on single node's audit data) detections. The *hierarchical architectures* amount to a multilayer approach, by dividing the network into clusters. Specific nodes are selected (based on specific criteria) to act as cluster-heads and undertake various responsibilities and roles in intrusion detection, which are usually different from those of the simple cluster members. The latter typically run a lightweight local intrusion detection engine that performs detection only on local audit data; while the cluster-heads run a more comprehensive engine that acts as a second layer of detection, based on audit data from all the cluster members. However, since the majority of the existing cooperative and hierarchical IDS architectures for MANETs are inherited from static or mobile networks, which differ radically from MANETs with respect to the network topology, available resources, nodes' mobility and security vulnerabilities, they present significant limitations and weaknesses, which are analyzed in section 2.

On the other hand, the intrusion detection engines employed in MANETS are classified into three main types (Mishra et al., 2004): (i) signature-based, (ii) anomaly-based, and (iii) specification-based. Signature-based engines rely on a predefined set of patterns (signatures) to identify attacks. The signatures are stored in a database and if the engine matches a monitored activity with a signature, then the activity is marked as malicious. This type of engines fails to detect novel attacks and requires always maintaining a signature database. The anomaly-based engines establish specific models of nodes' behaviors (normal profiles) and mark nodes that deviate from these profiles as malicious. This type of engines can detect unknown attacks and does not require a database. However, it is prone to high rates of false alarms, since any legitimate behavior that deviates from normal profiles is also considered as malicious. Finally, specification-based engines rely on a set of constrains or specifications that describe the correct operation of programs or protocols; and monitor the execution of programs/protocols with respect to the defined constraints/specifications. They combine the benefits of both signature and anomaly-based detection, since they: (i) can detect new types of attacks, (ii) do not maintain a database and (iii) do not present high rates of false alarms. However, the required constrains/specifications have to be manually developed, which might be time consuming.

This paper proposes a novel IDS for MANETs that aims at overcoming the limitations and weaknesses of the existing IDSs. The proposed IDS incorporates a novel random walk-based IDS architecture as well as a multi-layer, specification-based detection engine. The proposed solution consists of a set of self-contained Random Walk Detectors (RWDs), which randomly traverse a network, while monitoring each visiting node for malicious behaviors. The key advantage of this approach is that it is robust and scalable to network changes and produces little overhead. The number of RWDs on a network may increase and decrease accordingly, in order to cope with changes in the network topology, and, thus, RWDs may replicate or merge. At each visiting node, a RWD deploys a multi-layer, specification-based intrusion detection engine, which monitors the protocols and operations at the transport, network, and data-link layers, protecting the most critical functionality of MANETs. The proposed engine can detect both known and unknown attacks without requiring a database, and does not present high rates of false alarms.

The rest of this article is organized as follows. Section 2, outlines the limitations of the existing IDSs that motivate this work. Section 3, presents the proposed IDS, focusing on both the architecture and the detection engine. Section 4 briefly presents the advantages of the proposed solution as well as future work. Finally section 5 contains the conclusions.

2 MOTIVATION

2.1 Limitations of IDS Architectures

Taking into account the deployment environment of MANETs and its limitation, it is evident that the *processing overhead,* which is added by the IDS architectures to the underlying network nodes, should be kept to a minimum. However, in almost all the cooperative IDS architectures, one or more comprehensive detection engines (which are based on signature or anomaly detection) are employed in every node, without considering the limited processing capabilities. The hierarchical IDS architectures attempt to minimize the *processing overhead* by employing comprehensive or multi-layer detection engines only at some key nodes (i.e., cluster-heads), while the remaining nodes use lightweight engines. However, the creation and maintenance of clustered/hierarchical structures adds extra *processing* load to the network nodes. Moreover, in these architectures the relative high nodes' mobility, experienced in MANETs, may also increase the *processing* loads of the nodes.

In both cooperative and hierarchical IDS architectures, nodes have to exchange alerts, audit data, and detection results that impose extra *communication overhead* to the underlying network.

In the cooperative architectures, cooperation and the related overhead takes place only when a suspicious behavior cannot be resolved as malicious using only local audit data. On the other hand, in the hierarchical IDS architectures the *communication* overhead takes place when clustered/hierarchical structures are formed, a cluster-head is elected (or re-elected), the cluster members move and change clusters, or a cluster-head and the cluster-members exchange audit data. The hierarchical architectures also impose *unfair workload distribution among the network nodes,* since the nodes elected as cluster-heads are overloaded with detection responsibilities.

In both types of IDS architectures (i.e., cooperative and hierarchical) *nodes' mobility* decreases the *detection accuracy* and increases the *rate of false positives*. Mobility changes the network topology, the clusters' structure, the routing information maintained at each node, the created social and trusted relationships among the nodes, etc., influencing in that way the intrusion detection process. Moreover, a mobile node may move away from its neighboring nodes or from a detection engine that resides in a cluster-head, making cooperation for detection purposes or thorough inspection of the node unavailable.

Regarding security, the hierarchical IDS architectures present *points of failure,* since they place the responsibility of intrusion detection in a subset of elected nodes (i.e., cluster-heads). This fact makes these nodes potential targets of attacks, and if an attack succeeds then points of failure occur. Moreover, these architectures are *vulnerable to byzantine attacks*. Such an attack can take place during the election phase of a cluster-head, where a number of malicious nodes attempt to elect a malicious node as cluster-head. A malicious cluster-head may hinder intrusion detection or falsely accuse legitimate nodes as malicious. Another security weakness of both types of architectures (i.e., cooperative and hierarchical) is that they are exposed to *man in the middle* and *blackmail attacks.* Both of them rely on the exchange of intrusion detection information, either between cooperating nodes or between a cluster-head and the cluster-members, in order to perform detections. This information might be captured, modified, and retransmitted by a malicious node resulting in a *man in the middle attack*. Finally, a malicious node may transmit false information when requested upon by a cooperating neighbor or by a cluster-head, resulting in *a blackmail attack*.

2.2 Limitations of Detection Engines

Signature-based engines offer low detection latency and low rates of false positives, but they are not effective against new types of attacks, which are not included in the signatures database. Thus, administrators have to create up-to-date signatures in order to cope with new attacks. Furthermore, maintaining and updating a signature database in a MANET environment is difficult to achieve. Nodes in a MANET typically have limited memory, and a signature database requires a centralized signature distribution authority, which is contradictory to a MANET environment. Sterne et al. (Stern et al., 2005) have proposed a hierarchical scheme to distribute signatures in a MANET. However, this adds to the communication overhead, and each node has to allocate a specific memory portion to maintain the signature database. In addition, in this scheme, selfish nodes may block the signature distribution process or provide false signatures in order to hinder intrusion detection and impose damage to the network.

On the other hand, anomaly-based engines are the most popular for MANET IDSs, since they can detect unknown attacks and do not require a database. However, they rely on normal profiles, which might negatively affect the efficiency and performance of detection, in cases that dynamic changes in the network occur. More specifically, nodes' mobility changes the network topology and the routing information maintained at each node, resulting in high rates of false positives. Adjustable thresholds (Nadkarni et al., 2004) (Sun et al., 2007) try to reduce these negative impacts, since they ensure that periodical changes will remain under the detection threshold; while malicious behaviors that are persistent will exceed the thresholds indicating the occurrence of attacks. If for a preset period of time, no attack occurs, the threshold values are raised; otherwise they are lowered. However, the use of adjustable thresholds introduces new security weaknesses, since malicious nodes 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.

A specification-based engine compares, at run time, the behavior of objects (i.e., security-critical programs, protocols, or applications) with the associated security specifications. The latter are created based on the expected functional behavior of the objects. Therefore, a specification-based engine does not directly detect attacks, as happens in signature-based engines, but it detects the effects of them, as run-time violations of the specifications. As the engine relies on monitoring a set of specifications/constraints for breaches, instead of specific attacks, it can detect both known and unknown attacks. Moreover, it avoids the high rates of false alarms, since it does not use normal profiles, as happens in anomaly detection.

In general, the development of specifications for a specification-based engine might be a lengthy and convoluted process, since the developer has to determine what is the expected behavior of each individual application or protocol, and then establish constrains that characterize this behavior. However, in MANETs, this overhead can be reduced since the un-hindered operation of the network relies on specific protocols at the transport, network, and data-link layer, where the majority of security attacks occur (Djenouri et al., 2005) (Yang et al., 2004). Currently, specification-based engines for MANETs have limited use, as they monitor only the network layer for routing attacks (Tseng et al., 2003), (Tseng et al., 2005), (Hassan et al., 2006), (Huang et al., 2004), (Orset et al., 2005). In this paper, we extend the use of specification-based engines for MANETs that can monitor the transport, network, and data-link layers of the network stack, providing an integrated solution capable of detecting the majority of critical attacks.

3 THE PROPOSED IDS

3.1 IDS Architecture

The proposed IDS does not require the use of comprehensive detection engines at each network node, like the cooperative architectures, or any static structure like the hierarchical architectures. It consists of several robust RWDs that randomly traverse a network, while monitoring each visiting node for malicious behaviour. The number of RWDs on the network is scalable, in order to cope with changes in the network topology and thus RWDs may replicate or merge.

A Random Walker (RW) is a stochastic process, which represents a path of random successive steps. RWs can be applied to graphs, in which a RW process begins at a node on a graph and takes random successive steps to adjacent nodes. Thus, a RW can be seen as a method to randomly explore a graph (Lovasz, 1996). RWs provide a wide range of applications in computer science, physics, statistics, economics, and several other fields. In communication networks, RWs algorithms exhibit simplicity, low overhead, reliance only on local information, robustness to changes in a graph structure, and thus applications based on them are becoming more and more popular.

The two key advantages of RWs are: (i) they are inherently robust and scalable to network topology changes, since they do not require knowledge or state maintenance for the network structure; and (ii) they produce little overhead. For these reasons, they are particularly suitable in MANETs, where: (i) the network topology changes over time, since nodes move around the network bounds or join and leave dynamically without centralized control; and (ii) node resources are typically sparse. Therefore, the advantages of RWs can be used to address the previously mentioned limitations of the existing IDS architectures for MANETs. Currently, RWs find a plethora of applications in the context of MANETs, such as querying, service discovery, routing, service advertisement, searching, sampling, etc. (Kogias et al., 2008). However, to the best of our knowledge, they have not been proposed to support intrusion detection for MANETs.

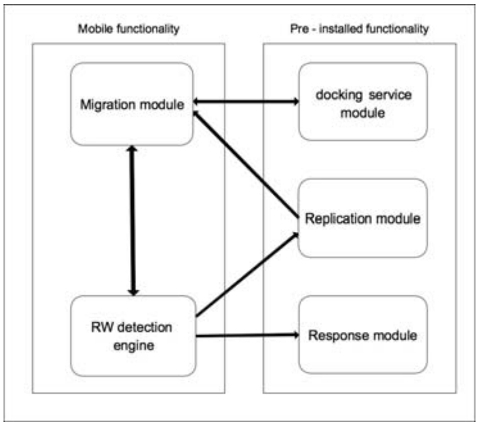

Figure 1: Layout of the RWD.

The proposed RWD is divided into five parts as illustrated in Figure 1: (i) the migration module that is responsible for the migration process of the RWD to a neighbouring node; (ii) the specification-based detection engine that includes the detection functionality of each RWD; (iii) the replication module that enables the RWD to be replicated; (iv) the response module that is responsible for notifying other nodes regarding malicious behaviours detected and for taking the required defensive action against them; and (v) the docking service module (which is executed in every network node) that monitors for incoming RWDs and is responsible for accepting and establishing a secure connection during the migration process. The replication, response, and docking modules are pre-installed in every node and utilized when a RWD visits that specific node. This approach alleviates the need for transmitting the full functionality of the IDS, thus reducing the communication overhead of the proposed architecture. On the other hand, the migration and detection modules are transferred during the RWD migration. This is because the first performs the migration process, while the second protects this process from attacks and verifies that the pre-installed modules have not been tampered. Subsequently, the functionality of each module of the proposed IDS is presented and analyzed.

3.1.1 Migration Module

The migration module of the RWD elects randomly a neighbouring node. Then, it establishes a secure communication channel with the docking service module of the node for the secure migration of the RWD. Security is achieved by using a symmetric key that minimizes the use of nodes' resources. In particular, AES (Daemen et al., 2002) is used for key generation, which consumes minimum battery resources for both key setup and encoding/decoding (Potlapally et al., 2006). To avoid the overhead of a key distribution scheme, key exchange can be achieved either through network steganography (Li et al., 2009) or elliptic curve Diffie-Helman (ECDH) (Miller, 1986) key exchange. Network steganography allows for the creation of covert channels, in which information is embedded within a variety of system properties and can only be detected by the designated user of the system. In wireless networks, information can be conveyed through MAC-layer covert channels (Li et al., 2009). On the other hand, ECDH is an asymmetric key exchange algorithm. It can be used to encrypt the symmetric key, with minimum energy cost (Potlapally et al., 2006) (i.e., compared to other public key algorithms), and without any cost on security (i.e., reduction of key size).

In general, the following steps take place during migration:

- **Key generation**: the migration module of the node that initiates migration generates a symmetric key, using AES.
- **Key exchange**: the key is transferred to the docking service module of the node selected for migration (using a covert channel or ECDH), and thus a secure channel between the two nodes is established.
- **RWD migration**: the RWD migrates to the selected node, through the newly established secure communication channel.

The process of migration is supervised by the detection engine, which monitors for any malicious activity on the part of the receiving node. If the pre-installed functionality at the receiving node is absent or tampered, the migration process is aborted and the receiving node is marked as malicious.

3.1.2 Detection Engine

Each RWD deploys a multi-layer, specification-based detection engine, analyzed in detail in section 3.2. Contrary to the existing collaborative architectures (i.e., cooperative and hierarchical), in which detection engines monitor nodes and decide on malicious behaviours, remotely, a RWD monitors a specific node when it visits it. As a result, the detection engine of a RWD is not necessary to gather audit data from neighbouring nodes and execute complex algorithms (i.e., anomaly detection) to detect abnormalities. Therefore, a simple, multi-layer, specification-based detection engine is suitable for this architecture.

The multi-layer specification-based detection engine has two main responsibilities: (i) to monitor the migration process of the RWD as mentioned previously; and (ii) to perform detection at the visited node. Therefore, it is called by the migration module just before the migration process, and just after the RWD has migrated to the node. During the migration process, the detection engine monitors for any denial of service (DoS) attack and executes a remote procedure call at the destination node. The latter performs a hash check at the pre-installed IDS modules of the destination node in order to determine whether the functionality of IDS exists and it has not been tampered.

After a successful migration, the detection engine begins monitoring the visited node, for a time $T_{monitoring}$, before the RWD migrates to another node. A prolonged stay of the RWD at a node increases the possibility of detecting an attack in it, at the cost of not detecting attacks at neighbouring nodes. The time $T_{monitoring}$ should be sufficient for the detection engine to detect possible attacks that take place in the visited node. Therefore, the RWD should stay longer in critical nodes, whose failure or malicious behaviour has larger impact on the network. Parameters that characterize the criticality /significance of a monitored node include: the number of neighbouring nodes, the number of connections served by the node, the number of packets traversing the node, etc. Additionally, $T_{monitoring}$ should be randomized to avoid its predictability, which might be exploited by adversaries. Following these, $T_{monitoring}$ is given by (1):

$$T_{monitoring} = (T_{\min} + T_{critical} + R) \quad (1)$$

T_{min} denotes the minimum time required by a RWD to detect possible attacks, $T_{critical}$ is the extra time added because of the criticality/significance of the monitored node, and R is a random time added in order to randomize $T_{monitoring}$. Finally, if a malicious behaviour is detected, the detection engine calls the response module, which is responsible for alerting other nodes and taking defensive actions against the attack.

3.1.3 Replication, Response and Docking Service Module

The replication module is responsible for selecting when a RWD will replicate, based on a probability *P*. To achieve the most optimal network coverage, a topology-dependent replication policy is used. In such policy, the probability of replication increases in dense network areas, allowing RWDs to cover different network paths. As the number of neighbouring nodes increases, the probability for replication increases exponentially. A generic replication probability is given by (2):

$$P(k_{RWD}) = -\left(e^{-k_{RWD}+1}\right) + 1 \quad (2),$$

where k_{RWD} is the number of neighbours of a node in which the RWD resides at. Having a low replication probability leads to scenarios where only a few RWDs traverse the network, increasing the time required to reach a malicious node and detect an attack (i.e., response time). On the other hand, having a high replication probability leads to flooding, where too many RWDs traverse the network. In future work, analytic and simulations studies will provide optimal values for the replication probability. When two or more RWDs visit the same node simultaneously, they merge in order to limit the amount of RWDs traversing the MANET and avoid redundant coverage at that particular network portion.

The response module is called by the detection engine when a malicious behaviour is detected. It is responsible for notifying the user(s) or administrator(s) for the detected behaviour and may take defensive actions against the attack, such as removing a malicious node from the routing table and notifying non-malicious nodes.

Finally, the docking service module is the only part of the proposed IDS that operates, continually, at every node on the network. It is executed as a system service and monitors for incoming RWDs. When a RWD attempts to migrate to a node, the node's docking service module is responsible for receiving a key and establishing a secure communication channel with the migration module of the node that the RWD originates from.

3.2 The Multi-layer Specification-based Detection Engine

The proposed detection engine performs detections using a set of specifications, which describe the normal node's operations at different layers, providing an aggregated solution. It monitors the most important protocols that provide end-to-end connectivity, routing, packet forwarding, and link layer connectivity. The advantage of this approach is twofold: (i) the overhead of specifications development can be reduced, since aggregated specifications are developed that focus on the three most important protocol layers; and (ii) the proposed multi-layer engine detects the majority of attacks (Djenouri et al., 2004) (Yang et al., 2004) that occur in MANETs, protecting the most critical/significant network operations.

In order to present the proposed engine, we use a Finite State Machine (FSM). Each state of the FSM corresponds to either a legitimate or malicious behaviour of the monitored node. A transition from one state to another is triggered by the node's operations/actions. Specifications are defined as a tuple (S, NO, S_0, δ, F) where S is the set of all possible states; NO is the set of node operations; S_0 is the initial state; δ is a function that maps node operations from a previous state to the current state; and F is the set of final states that correspond to malicious behaviours. The proposed multi-layer specification-based engine is a set of FSMs designed to monitor the correct operation of critical protocols at the transport, network, and data-link layers. To exemplify the operation of the engine, we illustrate a limited set of specifications, which enable the detection of some critical attacks, demonstrated in the following subsections. They cover the transport, network, and data-link layers and enable the detection of routing table poisoning, DoS, blackhole, impersonation, session hijacking, and SYN flooding attacks. In the following, for each layer, we analyze the specifications being monitored and outline the attacks detected.

3.2.1 Transport Layer Specifications

The transport layer protocols provide end-to-end connection, reliable packet delivery, flow control, congestion control, etc. The most important protocols at this layer are TCP and UDP used for connection-oriented and connectionless communication, respectively. Possible attacks that might be carried out at this layer include SYN flooding, session hijacking, UDP flooding, land attack, port scanning, man-in-the-middle, and spoofing. In Figure 2, we present a limited set of specifications used to supervise the correct establishment and operation of TCP connections at a node. It is well-known that a TCP connection is established through a three-way handshake. When the monitored node initiates a TCP connection, the detection engine should determine whether the node is attempting to establish a legitimate connection. To achieve this, it monitors whether: (i) the node encapsulates its legitimate address in the transmitted packets; and (ii) it acknowledges the three-way handshake. If the node attempts to encapsulate a false address or avoid transmitting an acknowledgment (ACK) packet, the engine reaches a state of malicious behaviour.

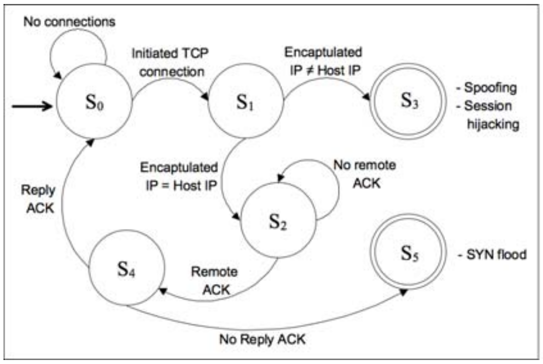

Figure 2: Transport layer specifications.

As illustrated in Figure 2, the engine remains in the initial state S_0, while no new TCP connection takes place. When a TCP connection is initiated, the engine moves to S_1. In this state, it verifies whether the monitored node encapsulates its actual address in the transmitted packets. If it attempts to encapsulate a different address, then the final state S_3 is reached, designating that the node is performing an impersonation (i.e., spoofing) attack. By transmitting a false address to a target node, the monitored node might also attempt a session hijacking attack, in which it impersonates a victim node and continues a session that was open between the victim and the target node. On the other hand, if the monitored node encapsulates its legitimate address, the engine moves to S_2. In this state, it monitors whether an ACK is received from the remote node. If such a packet is received, the engine moves to S_4. In this state the monitored node has to transmit an ACK packet to finalize the three-way handshake. If this happens, the engine returns to S_0; otherwise, it reaches the final state S_5, designating that the node attempts a SYN flood attack, since it does not complete the initiated TCP connection.

3.2.2 Network Layer Specifications

In MANETs, connectivity beyond one-hop neighbours is provided by routing protocols, which rely on the cooperation of all nodes. The most popular routing protocols for MANETs are the Ad-hoc On Demand Distance Vector (AODV) and the Dynamic Source Routing (DSR). Since both protocols are cooperative and distributed in nature, they are susceptible to a variety of attacks, which include wormholes, blackholes, byzantine, routing table overflow, routing table poisoning, rushing, packet fabrication, non-existing link advertisement, packet dropping, etc. Nevertheless, the majority of these attacks can be detected by monitoring the operation of the employed routing protocol.

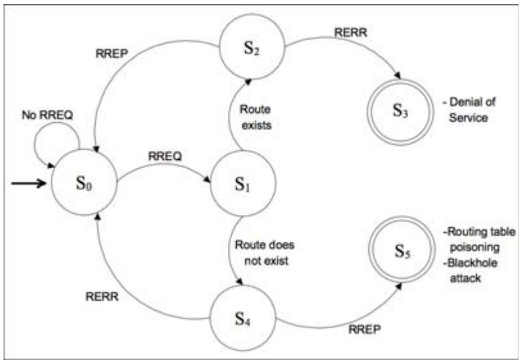

Figure 3: AODV monitoring specifications.

In Figure 3, we illustrate a limited set of specifications that monitor the AODV routing protocol, which establishes routes on demand. To ensure its correct operation, the engine supervises all route control messages at a node. When a node requires establishing a route to a destination node, it broadcasts a route request message (RREQ) to all of its neighbours. Nodes receiving the RREQ store a reverse route to the source node and forward the message. When the destination node receives the RREQ, it unicasts a route reply message (RREP) back to the source node. Intermediate nodes receiving the RREP store the route to the destination node in their routing tables. If the route to the destination node is broken, then a route error message (RERR) is transmitted back to the source node.

As presented in Figure 3, the detection engine awaits for incoming RREQ at the initial state S_0. When a RREQ is received, the engine moves to S_1 and observes the route validation process performed by the monitored node. If the requested route exists, the engine moves to S_2. In this state, the expected behaviour is to reply with a RREP. If this occurs, the route request process is completed and the engine returns to the initial state S_0. Otherwise, if the monitored node attempts to reply with a RERR message, the final state S_3 is reached, designating a DoS attack, since the node attempts to avoid participation in the routing process. On the other hand, if the requested route does not exist, the engine moves from S_1 to state S_4. In S_4, the legitimate behaviour of the monitored node would be to reply with a RERR message. If this happens, the engine returns to the initial state S_0. Otherwise, if the node attempts to transmit a RREP message, the final state S_5 is reached, designating a routing table poisoning or blackhole attack. In these attacks, the node misinforms other nodes regarding a non-existing route. Advertising such a route, the node attracts traffic in order to intercept packets. Then, it drops the packets without forwarding them.

3.2.2 Data-link Layer Specifications

The data-link layer is responsible for one-hop connectivity between neighbouring nodes. It consists of the Logical Link Control (LLC) and the Media Access Control (MAC) sub-layers. The IEEE 802.11 MAC protocol is a standard for MANETs, responsible for the coordination of transmissions on a common communication medium. It utilizes a distributed contention resolution mechanism for sharing the wireless channel among multiple wireless nodes. In this mechanism, when a node wants to transmit data, it initiates the process by sending a request to send (RTS) frame, and the destination node replies with a clear to send (CTS) frame. Any other node receiving the RTS or CTS frames retreats from transmitting any data for a certain time. The protocol is vulnerable to a variety of attacks such as DoS, traffic analysis, monitoring, MAC disruption, etc.

In Figure 4, we illustrate a limited set of specifications that facilitate the engine to monitor the 802.11 MAC for DoS attacks. It observes whether the monitored node has any data to transmit (i.e., state S_0). When this occurs, it moves to S_1 and monitors whether the communication channel is clear (i.e., the monitored node has not overheard an RTS/CTS, and the communication channel is not in use). If the later holds, the engine returns to the initial state S_0; otherwise, it moves to state S_2. In this state, the engine observes whether the monitored node attempts to use the occupied communication channel or retreats, until the RTS/CTS timer expires. If it attempts to transmit data, then the engine moves to the final state S_3, designating that the node is attempting a DoS attack. Otherwise, if there is no transmission within the RTS/CTS timeframe, the engine moves to the initial state S_0.

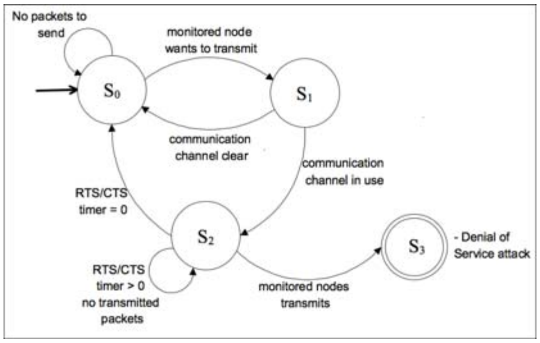
No packets to send
monitored node wants to transmit
S0
S1
communication channel clear
communication channel in use
RTS/CTS timer = 0
S2
S3
- Denial of Service attack
RTS/CTS timer > 0 no transmitted packets
monitored nodes transmits

Figure 4: 802.11 MAC protocol specifications

4 EVALUATION AND FUTURE WORK

The proposed RW-based IDS presents significant advantages compared to the existing IDSs for MANETs, which are briefly analysed bellow. The use of a specification-based engine enables the detection of known and unknown attacks, and alleviates the need for a signature database. In addition, the proposed detection engine is not prone to high rates of false alarms, as happens in anomaly-based detection, in cases that dynamic changes occur in the network (i.e., churn, changes in the topology, high node's mobility, etc.). A unique engine monitors the transport, network, and data-link layers, reducing the overhead typically associated with the development of specifications and facilitating the detection of most critical types of attacks. Current specification-based engines for MANETs focus only at the network layer and detect only routing attacks.

The proposed IDS architecture imposes less processing overhead than the stand-alone and cooperative architectures, since it does not require a comprehensive detection engine at each node.

Furthermore, the processing workload is uniformly distributed among the network nodes, since the RWDs are moved randomly. The communication load imposed by the movement (i.e., migration process) of RWDs consists of the volume of the executable code of the detection engine, while the remaining functionality is pre-installed on each network node. Thus, during the migration of a RWD only a small volume of data will be transmitted. Moreover, the communication overhead from the migration process on each link does not occur constantly, as happens in both cooperative and hierarchical architectures. The detection accuracy of the proposed IDS is not negatively affected by nodes' mobility, since detection does not rely on cooperation from other nodes or cluster members.

The proposed IDS does not create points of failure, since detection responsibilities are not concentrated to a specific node or a fixed set of nodes. A possible attack against one or more RWDs does not hinder the detection process in a network, since other RWDs traverse it. Moreover, the proposed IDS is not vulnerable to man-in-the-middle and blackmails attacks, since RWDs do not exchange audit data and the migration process of a RWD is protected through the use of an encrypted communication channel. Finally, since the detection tasks of a node are not assigned to other nodes, the proposed IDS does not enable malicious nodes to accuse legitimate nodes for malicious behaviour.

In future work, the proposed IDS will be evaluated through analytic and simulation studies and compared with existing IDSs. More specifically, the RW-based architecture will be evaluated using: (a) the response time to attacks, (b) the monitoring time of a node, and (c) the ratio of RWDs/nodes. By examining the response time of a RWD to attacks, we can assess the time period that nodes remain without protection when an attack takes place, and thus adjust the replication mechanism accordingly. The monitoring time of nodes (depends on nodes criticality/significance) exhibits the distribution of workload between nodes and is closely related to the detection accuracy, the ratio of false positives and the consumption of resources. Examining the ratio of RWD/nodes, we can assess the scalability of the proposed architecture in cases that the number of nodes increases or decreases. On the other hand, the specifications of the proposed engine will be further elaborated and enhanced to address the entire protocols employed at the transport, network, and data-link layers of MANETs. Moreover, the proposed engine will be evaluated regarding: (a) the provided detection accuracy, (b) the rate of false positives, and (c) the capability of detecting various attacks at multiple layers. Finally, we will evaluate the robustness of the proposed IDS under a variety of security attacks, and the level of security provided by the network steganography.

5 CONCLUSIONS

MANETs are susceptible to a variety of attacks that primarily target the protocols of the transport, network, and data-link layers. Currently, a large number of IDSs have been proposed that protect MANETs; however, the majority of them presents limitations and weaknesses, which mainly derive from the fact that they are inherited from static or mobile networks. This paper proposes a novel IDS that attempts to addresses the limitations and weaknesses of the existing IDSs. It includes a random walk-based architecture and a multi-layer, specification-based detection engine. The proposed architecture imposes less processing and communication overhead to the underlying network, it distributes uniformly the processing workload among the network nodes, and it is robust to dynamic network changes. Moreover, it does not create points of failure, and it is not vulnerable to man-in-the-middle and blackmail attacks. On the other hand, the proposed engine enables the detection of both known and unknown attacks, and alleviates the need for a signature database. Finally, it is not prone to high rates of false alarms.

REFERENCES

Mishra, A., Nadkarni, K., Patcha, A., 2004. Intrusion Detection in Wireless Ad Hoc Networks. IEEE Wireless Communications, Vol. 11, Issue 1, pp. 48-60.

Djenouri, D., Khelladi, L., Badache, N., 2005. A Survey of Security Issues in Mobile Ad Hoc Networks. IEEE Communications Surveys, Vol. 7, No. 4.

Yang, H., Luo, H., Ye, F., Lu. S., Zhang, L., 2004. Security in mobile ad hoc networks: challenges and solutions. IEEE Wireless Communications Surveys, Vol. 11, No 1, pp. 38–47.

Sen, S., Clark, J. A., 2009. Intrusion Detection in Mobile Ad Hoc Networks. Guide to Wireless Ad Hoc Networks, S. Misra, I. Woungang, S.C. Misra (Eds.), Springer, p. 427-454.

Sterne, D., Balasubramanyam, P., Carman, D., Wilson, B., Talpade, R., Ko, C., Balupari, R., Tseng, C-Y., Bowen, T., Levitt, K., Rowe, J., 2005. A General Cooperative Intrusion Detection Architecture for MANETs. Proceedings of the third IEEE

International Workshop on Information Assurance, pp. 57 – 70.

Nadkarni, K., Mishra, A., 2004. A Novel Intrusion Detection Approach for Wireless Ad Hoc Networks. IEEE Wireless Communications and Networking Conference (WCNC. 2004), vol. 2, pp. 831 – 836.

Sun, B., Wu, K., Xiao, Y., Wang, R., 2007. Integration of mobility and intrusion detection for wireless ad hoc networks. International Journal of Communication Systems, vol. 20, Issue 6, pp. 695 – 721.

Lovasz, L., 1996. Random walks on graphs: a survey. Combinatorics: Paul Erdos is eighty (Keszthely, Hungary, 1993), vol. 2, edited by D. Miklos et al., Bolyai Soc. Math. Stud. 2, J´ anos Bolyai Math. Soc., pp. 353–397.

Kogias, D., Oikonomou, K., Stavrakakis, I., 2008. Replicated Random Walks for Service Advertising in Unstructured Environments", to appear in the 7th IFIP Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net), 23-27, Palma de Mallorca, Spain.

Daemen, J., Rijmen, V., 2002. The Design of Rijndael, AES - The Advanced Encryption Standard. Springer Verlag, pp. 238.

Miller, V., 1986. Uses of Elliptic Curves in Cryptography," Proceedings of Crypto '85, LNCS 218, Springer-Verlag, pp. 417-426.

Li, S., Ephremides, A., 2009. Covert Channels in Ad-Hoc Wireless Networks. Elsevier Ad Hoc Networks.

Tseng. C.-Y., et al., 2003. A specification-based intrusion detection system for AODV. In proceedings. of ACM Workshop on Security of ad hoc and sensor networks.

Tseng, C. H., Song, T., Balasubramanyam, P., Ko, C., Levitt, K., 2005. A Specification-based Intrusion Detection Model for OLSR. In proceedings of the 8th International Symposium, RAID 2005, Recent Advances in Intrusion M., El-Kassas, S., 2006. Securing the AODV protocol using specification-based intrusion detection. In proceedings of the 2nd ACM international workshop on quality of service & security for wireless and mobile networks, Terromolinos, Spain.

Huang, Y., Lee, W., 2004. Attack analysis and detection for ad hoc routing protocols. In proceedings of the 7th International Symposium on Recent Advances in Intrusion Detection (RAID'04).

Orset, J., Alcalde, B., Cavalli,A., 2005. An EFSM-based intrusion detection system for ad hoc networks. In proceedings of the 3rd international symposium on Automated technology for verification and analysis, (ATVA 2005),Taipei, Taiwan.

Potlapally, N. R., Ravi, S., Raghunathan, A., Jha, N. K., 2006. A Study of the Energy Consumption Characteristics of Cryptographic Algorithms and Security Protocols. IEEE Transactions on Mobile Computing, v.5 n.2, p.128-143.