Weak Process Models for Attack Detection in a Clustered Sensor Network Using Mobile Agents

Marco Pugliese[1], Annarita Giani[2], and Fortunato Santucci[1]

[1] Center of Excellence DEWS, University of L'Aquila, L'Aquila, Italy
marco.pugliese@ieee.org, santucci@ing.univaq.it
[2] Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, Berkeley, CA, USA
agiani@eecs.berkeley.edu

Abstract. This paper proposes a methodology for detecting network-layer anomalies in wireless sensor networks using weak process models (WPM). Weak process models are a non-parametric version of Hidden Markov models (HMM), wherein state transition probabilities are reduced to rules of reachability. Specifically, we present an intrusion detection system based on anomaly detection logic. It identifies any observable event correlated to a threat by applying a set of anomaly rules to the incoming traffic. Attacks are classified into low and high potential attacks according to its final state. Alarms are issued as soon as one or more high potential attacks are detected.

We model hello flooding, sinkhole and wormhole. We introduced single threat models and aggregated models and study how effective they are to detect each attack.

We present the design approach for the proposed WPM-based detection technique using mobile agents. Early implementations of the agent based secure platform have already been implemented.

Keywords: Weak Process Models, Anomaly Detection, Threat Identification, Alarm Generation.

1 Introduction

Sensor networks permit data collection and computation to be deeply embedded in the physical environment. Sensor nodes are often left unattended so that they are susceptible to security attacks. Typical threats affecting wireless sensor networks (WSNs) are reported in [1]. An intrusion detection system (IDS) is a defense system, which detects hostile network activities. It recognizes patterns of known attacks (signature based) [19,18], or identifies network activities that differ from historical norms (anomaly based) [2].

This work considers an IDS based on anomaly detection logic (ADL). Threats are correlated to any sequences of observable events by applying a set of anomaly rules to the incoming traffic. Computer networks are typically provided with mechanisms to identify changes in system parameters or anomalous exchange of

S. Hailes, S. Sicari, and G. Roussos (Eds.): S-Cube 2009, LNICST 24, pp. 33–50, 2009.

information. Such data can be used as relevant observations to predict the hidden state of the system and infer if it is under attack. A HMM is a doubly stochastic finite state machine with an underlying stochastic process that represents the real state of the system. The real state of the system is hidden but indirectly observable through another stochastic process that produce a sequence of observable events [5,15]. The relationships between hidden states and observable data are stochastic as well as the transitions between states. HMMs [5,15] have been widely used in network-based IDS for wired systems [9,10,11,12,22,23] as well as for modeling Internet traffic [21]. The Baum-Welch algorithm as likelihood criterion and parameter estimation is extensively used [5].

Silva et al. [13] have proposed a decentralized IDS that fits demands and restrictions of WSNs. The network behavior is obtained from the analysis of events detected by a monitor node. A set of rules are compared with the information gathered from the network traffic. In general, application of traditional IDSs to sensor networks is challenging. In fact they require intense computation capability [20] and they are too limited to a restricted number of threats [7]. Some conventional intrusion detection systems perform cross-correlation and aggregation of data. For example, they analyze fluctuation in sensor readings [6], or detect abnormal traffic patterns [26].

Implementing an effective IDS on a wireless sensor network leads to the problem of finding a trade-off between the capability of identifying threats (i.e. with a bounded false alarm rate), the complexity of the algorithms and memory usage [8]. Doumit and Agrawal [7] proposed a novel approach for applying a lightweight, yet robust IDS designed for wireless sensor networks based on self-organized criticality and HMM. They model the natural dynamic of the system so that unusual activity can be identified. We propose here an intrusion detection system which replaces HMMs with WPMs. WPMs are a non-parametric version of HMMs wherein state transition probabilities are reduced to rules of reachability.

Very low state transition probabilities are reduced to zero which increases false negatives. This means that some sequences are classified as not possible when instead in a probabilistic model would be achievable. The number of false negatives decreases if we add states [4] but the drawback is a larger memory requirement. The matrices that describe the models are sparse and can be compacted for faster computation.

The estimation of a threat in the case of weak processes is greatly simplified and less demanding for resources. The *most probable* state sequence generated by the Viterbi algorithm [3] for HMM becomes the *possible* state sequence. The intensity of the attack is evaluated by introducing a threat score, a likelihood criterion based on weighting states and transitions [4]. Intrusions and violations are classified into low potential attacks (LPA) and high potential attacks (HPA) depending on their distance from the state corresponding to a successful attack. When at least one HPA occurs, an alarm is issued.

We assume that a secure routing protocol [14,24] is running on the network and that routing messages are ciphered and authenticated through an underlying

cryptographic scheme. We analyze hello flooding, sinkhole and wormhole. A secure routing protocol would result not effective to protect from these threats [2,1]. According to these assumptions, we can mainly focus on threats from internal intruders that generate control messages that are syntactically and semantically well-formed.

In summary, the main contributions of this paper are as follows:

- Application of our method to the detection of sinkhole and wormhole (hello flooding was considered in [4]);
- Extension to models that describe aggregate threats;
- Comparison of models that describe a single attack and models that describe more than one attack in terms of detection capability.

The remainder of this paper is organized as follows. In section 2 we present some background material on weak process models and show how it is applied to the problem of detecting sinkhole and wormhole. In section 3 we analyze false alarms and misdetection of single threat models compared to aggregated models. In section 4 we will present the design approach for the proposed WPM-based detection technique: mobile agents and enhancements to the execution environment in [31] have been proposed. Early implementations of the agent based secure platform are already available in our lab. Section 5 contains concluding discussion and future work.

2 Threat Detection and Alarm Generation

In our anomaly-based approach the anomaly rules are defined through inequality so that they define regions of the state space. This allows us to introduce a ranking among states that leads to a hierarchical structure. The number of false negative (mis-detection) is reduced since inequalities are satisfied by a larger number of values with respect to equalities. Also the number of false positive decreases given that we choose to associate an alert only to the state with the highest risk. States are classified according to two hazard levels, low potential attack (LPA) states and high potential attack (HPA) states. We also introduce a score mechanism to weight state sequences where LPA and HPA states contribute differently. Now we give formal definitions of WPMs, threat score, low potential attack and high potential attack.

A WPM, as any Markov model, can be formally represented using the canonical form:

$$\begin{cases} x^{k+1} = Ax^k \\ o^k = Bx^k \end{cases} \tag{1}$$

where:

- $X = (x_1, x_2, \ldots, x_3)$ *is the state set*;
- x^k is the *state* at step k. x^0 is the *initial state*;

- $O = (o_1, o_2, \ldots, o_q)$ is the *the set of observations*;
- o^k is the *observable* o at step k ;
- A is a $n \times n$ matrix representing *state transition distribution*. Matrix elements are defined as:

$$A_{i,j} = \begin{cases} 1 & \text{if } p(x^{k+1} = x_j | x^k = x_i) = 1 \\ 0 & \text{otherwise} \end{cases}$$

- B is a $q \times n$ matrix representing the *emission distribution*, that maps each observable with the states. Matrix elements are defined as

$$B_{i,j} = \begin{cases} 1 & \text{if } p(o^k = o_j | x^k = x_i) = 1 \\ 0 & \text{otherwise} \end{cases}$$

Definition 1. *Let s^k be the threat score after k observations. s^k is the result of a weighting mechanism applied to states and transitions belonging to the hypothetic state trace. Weights are represented by a square n* x *n matrix S, where is n the number of states in the model. The elements of the matrix are defined as follows:*

$$s_{ij} = \begin{cases} \text{weight assigned to the transition from } \mathrm{x_i} \text{ to } \mathrm{x_j} & \text{if } i \neq j \\ \text{weight assigned to the state } \mathrm{x_i} & \text{if } i = j \end{cases}$$

Definition 2. *A Low Potential Attack (LPA) is an attack defined by a state x_j whose distance from the final state is at least 2 hops.*

Definition 3. *A High Potential Attack (HPA) is an attack defined by a state x_j whose distance from the final state is less than 2 hops.*

Let us assume now that, if a node represents a LPA state, then its score is L. If a node represents a HPA state, then its score is H and, if a node neither represents an attack state nor is the final state, then its score is 0. L and H are integers. We define the elements of the score matrix S as follows:

$$s_{ij} = \begin{cases} a_{i,j} \cdot (s_i - s_j) & \text{if } i \neq j \\ \{0, L, H\} & \text{if } i = j \end{cases}$$

where a_{ij} are the elements of matrix associated to the model describing the threat.

Let us define n^k_{hpa} and n^k_{lpa} the numbers of high potential and low potential states, respectively, reached in the observation interval. These numbers are not limited to a single trace, but include states belonging to all possible state sequences at time k. With the above assumptions the threat score at time k is

$$s^k = H \cdot n^k_{hpa} + L \cdot n^k_{lpa} \tag{2}$$

The choice of H and L depends on the length of the memory that we allocated to store the various states of the model. We call it WML, the weak process model

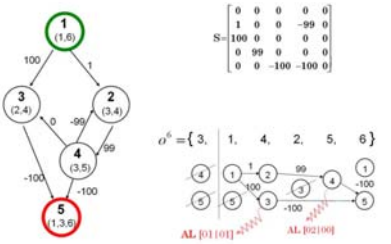

Fig. 1. Weak process model generating two alarms

length. If n is the number of states, then $n \cdot WML$ is the number of states to store. The worst case is when a single observable is associated to all $n \cdot WML$ states so that $\frac{H}{L} \geq n \cdot WML$.

In Figure 1 we show a model, the score matrix and the sequence of observations emitted at each step. The observable sequence is $o^6 = \{3, 1, 4, 2, 5, 6\}$. We assumed L=1 and H=100. At the bottom right is the correspondence between observables and associated states. The first observation must be discarded because the starting state can be reached only in the second observation. The two possible state traces are $Tr_1^6 = 1, 2, 4, 5$ and $Tr_2^6 = 1, 3, 5$. When the attack reaches state 3 or state 4 an alarm is issued (they are HPAs). Let us now compute the scores in state 3 and state 4. The state number 3 is reached with the third observation. The sequence of hidden states until the third observation has 1 high potential state (state 3) and 1 low potential state (state 1). Therefore, the score at state 3 is

$$s^3 = H \cdot n_{hpa}^3 + L \cdot n_{lpa}^3 = 100 \cdot 1 + 1 \cdot 1 = 101$$

The state number 4 is reached with the fifth observation. The sequence of hidden states until the fifth observation has 1 high potential state (state 4). We also have to consider the state (3). So we have two HPA states. This gives us the score at state 4

$$s^4 = H \cdot n_{hpa}^4 + L \cdot n_{lpa}^4 = 100 \cdot (1 + 1) + 1 \cdot (0) = 200$$

A cluster is a group of nodes that are interconnected. A dedicated node of the cluster is called cluster head. The cluster head is responsible for scheduling and dissemination of messages to the cluster members and for data aggregation when necessary [30]. In the rest of this paper we will concentrate on models with a minimal clustered topology composed by only three nodes: a cluster head CH, the generic cluster member M_i, and the attacking node n_e (figure 3).

An anomaly rule is a logic filter applied to incoming messages. If the filtering results in absence of anomalies, the message is processed further; otherwise, if an anomaly is detected, we are in the case of a threat. Any rule can be applied indifferently either to the cluster head and to the members as well. This scheme is scalable and avoids rule explosion.

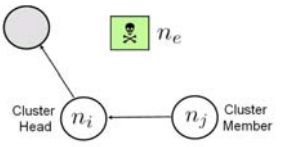

Fig. 2. Model with a minimal clustered topology

We follow a four step process when modeling a threat.

1. Analyze the behaviour of the threat;
2. Derive the Anomaly Rules;
3. Derive the WPM-based threat model;
4. Assign weights to WPM states and transitions.

Given the set of threat observables collected during two consecutive observation steps, the observable with the highest score back-propagates. In [4] we describe how to model the hello flooding attack and how WPMs are used to detect it. In this work we concentrate instead on the sinkhole and wormhole attack modeling and detection.

In the following threat scenarios, dotted arrows indicate the malicious traffic flow and the label refers to the anomaly rule used to detect it.

2.1 Sinkhole Modeling and Detection

In this threat nearly all traffic from a particular area is lured through a specific compromised node with unfaithful routing information [14]. Each neighbouring node of the adversary node is induced to forward packets directed to a base station through the adversary. The malicious node can then suppress, modify or redirect the packets. Geographic routing protocol are resistant to this threat since traffic is routed based on physical location. Protocols that construct a topology initiated by a base station are most susceptible to wormhole and sinkhole attacks. Those protocols that construct a topology on demand using only localized interactions and information are more resistant to these attacks. In [27] a light-weighted algorithm is proposed for detecting sinkhole attacks. It assumes a base station centric approach for network flow collection and intrusion detection. In [28] a distributed IDS is introduced using MintRoute as underlying not secured routing protocol (widely implemented in TinyOS [16]). The importance of monitoring the hop-count parameter in order to detect sinkhole attacks is presented in [29]. The authors also present a computationally efficient scheme for detecting abnormal route advertisements.

We assume that the attack is highly hazardous if at least 2 nodes in the network are attacked. The numeric labels that appears in Figure 3 refers to the corresponding observables listed in Table 1.

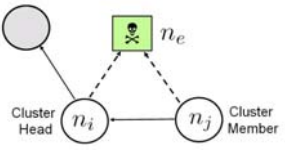

Fig. 3. Sinkhole attack. The attacker induces cluster members n_i and n_j to redirect messages to it.

Table 1. Anomaly rules description for sinkhole

ID	
AR1	If node n_i has authenticated node n_e but node n_e declares that $h_e < h_i$ with $h_i \neq 0$ (n_e declares it is the new cluster head of n_i but it is not) then $o^k = o_1$
AR2	If node n_i is the cluster head and (rule AR1 or rule AR2) in n_j is true then $o^k = o_2$

We introduce a counter R that is set to zero every time a new observation arrives. This is important to detect when observations are not arriving for more than a certain number of steps.

We indicate the malicious node with n_e, as evil. The attack is transparent if the compromised node n_e does not generate any anomalies and nodes n_i and n_j believe that they are supposed to connect with node e when instead their current cluster heads are alive. In [17] we stated the quantitative conditions among nodes hop distances h from the true sink to detect this anomaly: e.g. if node n_i is the cluster head of node n_j then $h_i = h_j - 1$ holds, where h_i and h_j are the respective distance hops from the sink.

The WPM-based sinkhole is represented in Figure 4. It has 4 states and 3 observables. The threat starts if observable 1 or 2 occurs and state SH_1 defines a LPA. If no more observables are identified in the following K steps (with K predefined threshold) then the threat is considered "reset", which means either that the attack is temporary suspended or there were no attack at all (SH_3). If either the observable 1 or 2 occurs again then the attack is dangerous. State SH_2 is high potential and an alarm is issued. If no more observables are identified in the following K steps then the attack is reset. The final state SH_4, labelled SUCCESSFULL ATTACK, is never reached so that the alarm remains on until an appropriate countermeasure has been taken or the threat returns reset.

The canonical form (1) and the score matrix can be specialized using matrices A_{SH} and B_{SH} and S_{SH} in (3).

$$A_{SH} = \begin{bmatrix} 0\,0\,0\,0 \\ 1\,0\,0\,0 \\ 1\,1\,0\,0 \\ 0\,1\,0\,0 \end{bmatrix} \qquad B_{SH} = \begin{bmatrix} 1\,1\,0\,0 \\ 1\,1\,0\,0 \\ 0\,0\,1\,0 \end{bmatrix} \qquad S_{SH} = \begin{bmatrix} 1 & 0 & 0\,0 \\ 99 & -100 & 0\,0 \\ -1 & 0 & 0\,0 \\ 0 & 0 & 0\,0 \end{bmatrix} \tag{3}$$

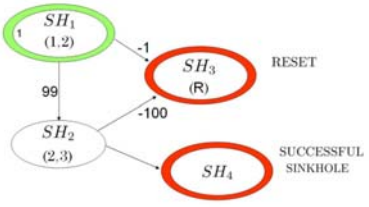

Fig. 4. WPM-based sinkhole model

2.2 Wormhole Modeling and Detection

In a wormhole attack a malicious node receives packets at one location in the network and tunnels them to another location (low latency link) in the network, where the packets are resent into the network generating denial of service, waste of resource, alteration on information semantics and other damages. In [25] wormhole attacks are detected by introducing the notion of a packet leash: a leash is any information that is added to a packet and is designed to restrict the packet's maximum allowed transmission distance. The authors distinguish between geographical leashes and temporal leashes: a geographical leash ensures that the recipient of the packet is within a certain distance from the sender; a temporal leash ensures that the packet has an upper bound on its lifetime, which restricts the maximum travel distance. Either type of leash can prevent the wormhole attack, because it allows the receiver of a packet to detect if the packet traveled further than the leash allows.

Figure 5 and Figure 6 depict two possible scenarios for wormhole attacks, where tunnel node end-points are located in the same cluster (intra-cluster attack scenario) or in different clusters (inter-cluster attack scenario) respectively. The

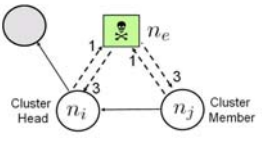

Fig. 5. Wormhole against nodes belonging to a cluster. The numeric labels refer to the corresponding observables listed in Table 2.

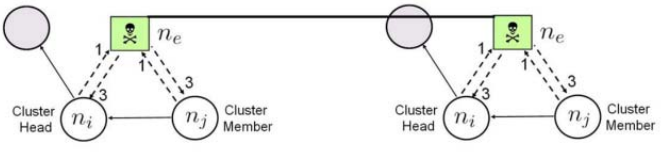

Fig. 6. Wormhole against nodes belonging to two different clusters

attacker emulates to be cluster head in order to receive traffic and to be cluster member to resend traffic or vice-versa. For example, in Figure 5 the sequence 1 - 3 (corresponding to the observable sequence $o_1, \ldots, o_3$) indicates a potential wormhole.

An informal description of the anomaly rules is provided in Table 2.

Observation o_R is emitted when no observables are detected for a certain number of steps. The last two anomaly rules, 3 and 4, describe a request to join a network. This can be an ordinary procedure as well as the precondition for a security attack. Therefore we suggest that the behavior related to AR3 and AR4 were consider normal and abnormal simultaneously. We retain as *ambiguous* threat observables with this characteristic. AR2 and AR4 enable the generation of the back propagation of the observables to the sink. This is important to detect complex threats attacking nodes that are distant to each other.

The WPM-based wormhole is represented in figure 7. The same considerations about RESET and SUCCESSFULLY ATTACK states made for sinkhole apply to this case. The number of states is 6 and the number of observables is 5. It is important to note that observables for sinkhole are a sub-set of those for wormhole. The canonical form (1) can be specialized using matrices A_{WH} and B_{WH} and the score matrix S_{WH} in eq. (4).

Table 2. Anomaly rules description for wormhole

ID	
AR1	If node n_i has authenticated node n_e but node n_e declares $h_e < h_i$ with $h_i \neq 0$ (n_e declares it is the new cluster head but it is not) then $o^k = o_1$
AR2	If node n_i is the cluster head and (rule AR1 or rule AR2) applied to n_j is true then $o^k = o_2$
AR3	If node n_i has authenticated node n_e but node n_E declare that $h_e \geq h_i$ with $h_i \neq 0$ (n_e declares it is the new cluster member but it is not) then $o^k = o_3$
AR4	If node n_i is the cluster head and (rule AR3 or rule AR4) applied to n_j is true then $o^k = o_4$

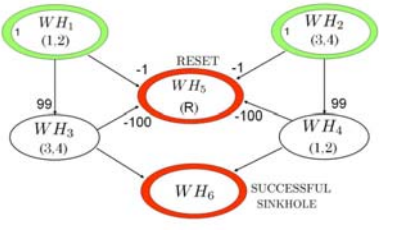

Fig. 7. WPM-based wormhole model

The threat starts with observables o_1 or o_2 or o_3 or o_4. If no more observables are identified in the next K steps the threat is considered suspended. If observables o_1 or o_2 or o_3 or o_4 occur again, then the attack moves to a high potential state.

$$A_{WH} = \begin{bmatrix} 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 1&0&0&0&0&0 \\ 0&1&0&0&0&0 \\ 1&1&1&1&0&0 \\ 0&0&1&1&0&0 \end{bmatrix} \quad B_{WH} = \begin{bmatrix} 1&0&0&1&0&0 \\ 1&0&0&1&0&0 \\ 1&1&1&0&0&0 \\ 0&1&1&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&0&0&1&0 \end{bmatrix} \quad S_{WH} = \begin{bmatrix} 1&0&0&0&0&0 \\ 0&1&0&0&0&0 \\ 99&0&0&0&0&0 \\ 0&99&0&0&0&0 \\ -1&-1&-100&-100&0&0 \\ 0&0&1&1&0&0 \end{bmatrix} \tag{4}$$

2.3 Aggregated Models

Single threat models can be aggregated in a multi threat model using boolean OR. The complete list of anomaly rules and the produced observables is the collection of the anomaly rules and observables related to single threats. We assume that cluster heads aggregate only the observable coming from the cluster members that has the highest score.

The aggregated threat model of hello flooding, sinkhole and a wormhole attack is shown in figure 8. When no more observables are identified in K consecutive observation steps the RESET state is reached. In that case the attack is suspended or there was no attack at all. Aggregated models allow to detect a larger class of attacks compared to single models.

In the next section we will show how the IDS proposed behaves in terms of false negative (mis-detection) or false positive (false alarms). The experiment investigates the accuracy of threat identification for single models and aggregated models.

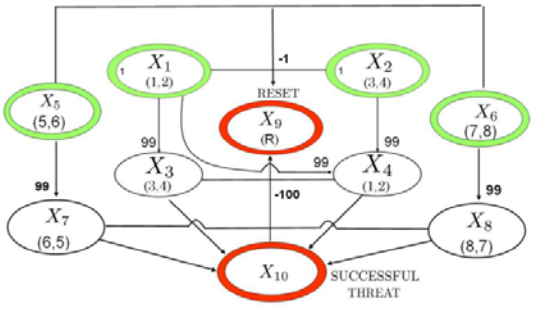

Fig. 8. Aggregated threat model for hello flooding, sinkhole and wormhole

3 Mis-Detection and False Alarm

It is important to check that if a threat occurs than an alarm is generated and that the number of false alarms is low. We investigate mis-detection and false alarm in the case of aggregated models in comparison to single threat models.

3.1 Mis-Detection Analysis

We apply the observable sequences produced by single hello flooding, sinkhole or wormhole. Each sequence will be applied to both single and aggregated model and their detection capabilities will be investigated. We assume 32 observables per sequence, $n \cdot WML \leq 100$ and $k = 3$. We consider the following observation sequences generated in the case of hello flooding, sinkhole and wormhole respectively.

$$\{5, 5, *, 8, 7, *, *, 6, 6, 8, 8, *, *, *, 8, *, *, 5, 7, *, *, 7, *, 6, 8, *, *, *, 5, 5, *, *\} \quad (5)$$

$$\{2, 1, *, *, *, 1, *, 1, 2, 2, *, *, *, 1, 2, *, *, 1, 2, *, *, *, 2, 2, *, *, 1, *, 1, *, *, *\} \quad (6)$$

$$\{2, 4, *, *, *, 3, *, 3, 1, 2, 2, *, *, 1, 3, *, *, 4, 4, *, *, *, 2, 2, *, *, 4, *, 3, *, *, *\} \quad (7)$$

It is important to note that there is no overlapping between observables from hello flooding and observables from sinkhole or wormhole. And the observables from sinkhole are observables from wormhole too.

Simulations provide the results graphically reported in figure 9, 10, and figure 11 respectively. Red bars refer to scores produced by individual threat models while yellow bars refer to aggregated threat models. To be noted that scores go to zero when the system is in a RESET state.

As expected the same outputs single model and aggregated model are obtained for threats not sharing any threat observables with each other. In fact figure 9 and figure 10 show that the aggregate and the single models have the same detection capability in cases of hello flooding and sinkhole respectively. Different

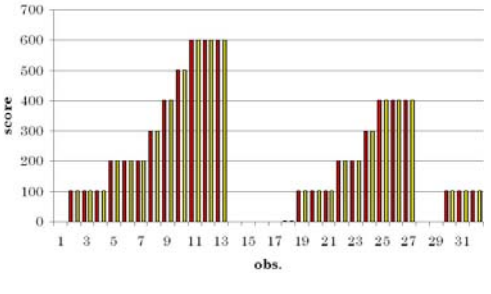

Fig. 9. Hello flooding detection scores when (5) is applied as input

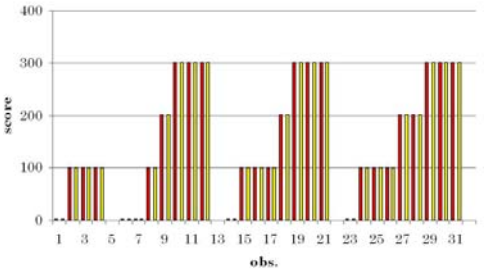

Fig. 10. Sinkhole detection scores when (6) is applied as input

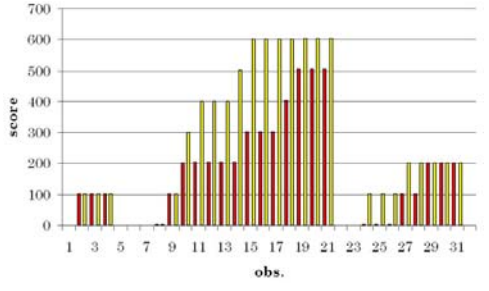

Fig. 11. Wormhole detection scores when (7) is applied as input

outputs are obtained for threats sharing at least one threat observables. For example sinkhole and wormhole in figure 10 and 11 share threat observables 1 and 2. In particular a sinkhole is an attack or is the initial step of a wormhole. Accordingly the score for wormhole in the case of the aggregated model is always greater than the score resulting from the single threat model (figure 11).

Attacks against distant nodes. Now we examine the capability of detecting the same threat attacking nodes that are not close to each other. In this test we assume a wormhole against nodes 4 and 5 in Figure 12. We consider the test successful if alarms were generated also in nodes 1, 2 and 3 (not directly attacked).

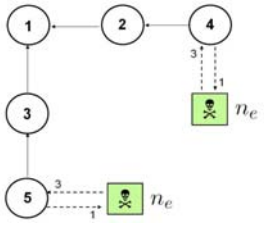

Fig. 12. Wormhole attack against nodes 4 and 5

Rule AR2, AR4 (from wormhole and the aggregate model) allow observable back propagation to nodes 4-2-1 and to nodes 5-3-1.

The criterion for the aggregation of observables is as follows. At the generic observation step k, the cluster head CH considers only observables with the highest score from neighbors. Accordingly, if the observable sequence 7 is produced in nodes 4 and 5, the following observables sequence 8 will be produced in nodes 1, 2 and 3:

$$\{2, 4, *, *, *, 4, *, 4, 4, 2, 4, *, *, 2, 4, *, *, 4, 4, *, *, *, 2, 2, *, *, 4, *, 4, *, *, *\} \quad (8)$$

Figure 13 shows the related threat scores. If no countermeasures are applied to nodes 4 and 5 observables propagates back from the compromised node to the sink until an alarm is triggered.

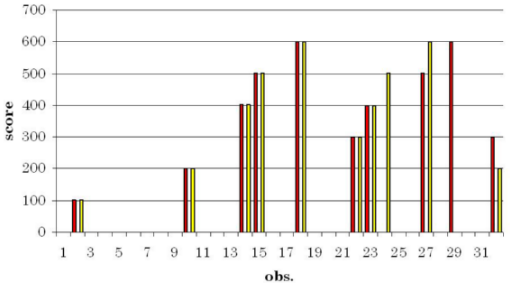

Fig. 13. Scores in nodes 1, 2, 3 from wormhole attack to nodes 4 and 5

3.2 False Alarm Detection

Test for positives will be performed through the analysis of the structure of the anomaly rules and the internal structure of the aggregated threat model in figure 8.

Among the anomaly rules in Table 2, AR3 and AR4 are associated to observables that can potentially produce false positives. This can lead to false alarms. But alarms be triggered only if the observations generate is a high potential attack. For example in figure 8, state X_3 is associated to observation o_3 or o_4 produced by AR3 and AR4 respectively. We propose two approaches reduce false positives.

1. *Introducing further states associated to certain threat observables in paths where at least one state is associated to ambiguous threat observables.* This approach lowers the probability for false positives ($p \mapsto 0$), as the longer the path to HPA the more reliable would be an alarm. A drawback is that long paths to HPA states, would reduce the reactivity in the monitoring service.
2. *Introducing a further class of states associated to ambiguous threat observables.* This approach cannot lower the probability for false positives, but

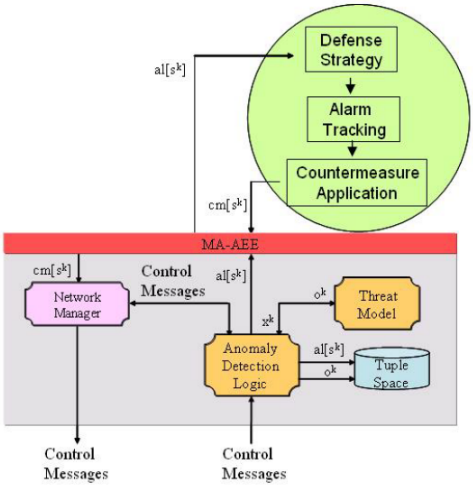

Fig. 14. WPM-based IDS component-based internal structure

"ad-hoc" lighter countermeasures can be applied to nodes where alarms from dubious observables are generated (e.g. node quarantine rather than link release).

4 Mobile Agent-Based Design

We adopt a mobile agent architecture to design and develop the proposed WPM-based intrusion detection. Resource constraints and topology dynamics in WSN imply restrictions for the software architectural choices where fundamental requirements are distribution, flexibility and scalability.

We will show that the agent based middleware proposed in [31] optimizes the design of distributed applications on clustered sensor networks. Cluster heads are not permanently assigned to specific nodes but dynamically re-assigned to any node according to eligibility criteria. This justify our choice of agents. A key concept from [31] is agent migration. During cloning, it copies its code and state to another node (strong cloning) and resumes executing on both the old and new nodes. Mobile agents support data-centric applications. Code migrates towards data independently from node addressing. Applications distribution through mobile agents results much less costly compared to traditional data broadcasting or code diffusion approaches. From [31] we will recall the concept of 'tuple space' as a local memory shared by local agents.

Figure 14 shows the basic functionalities of our intrusion detection system. Other functions related to intrusion reaction logic, including defence strategy, alarm tracking and countermeasures will be mapped into mobile agents as these functions will be performed at cluster level and data aggregation from neighbor nodes is mandatory. We denote this agent as 'Intrusion Reaction Agent' (IRA),

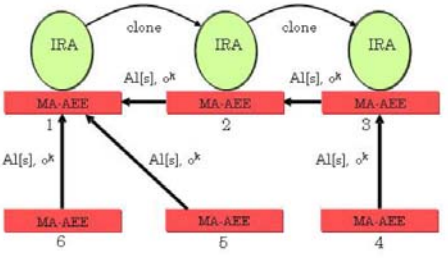

Fig. 15. IRA agent forward propagation vs. threat observables and alarms back propagation

represented by the oval in figure 14, which is hosted only by cluster heads. The mobile agent mechanism leads to forward propagation or diffusion from sink to leaves of IRA agents and to a back-propagation (from leaves to sink) of threat observables and alarms. In a clustered network, if IRA agent are hosted only on a cluster head, the agent diffusion mechanism across the network is described in figure 15 . IRA agent is initially hosted on node 1 and node 2 is a cluster member. The cluster head can read remotely threat observables stored on the tuple space of the neighboring node 2. When node 2 becomes cluster head, IRA (weakly) clones to it. The final agent distribution is depicted in figure 15. This mechanism allows each IRA to aggregate threat observables and alarms from cluster nodes, generate iteratively further observables and alarm. This aggregate and back-propagate mechanism for alarms and observables, leads to the detection of organized threats attacking nodes topologically distant. The ADL will store the produced observables and alarms into the 'local tuple' space. The middleware supporting the mobile agent execution environment is denoted as Mobile Agent Application Execution Environment (MA-AEE).

Our contribution is the definite verticality respect to WSN technology. Our approach to detection is fully distributed with a dynamic hierarchical architecture rather than centralized with a static hierarchical architecture. Security functions are executed autonomously by nodes in the network without any support from outside (like servers or database) as in [32,33], and complexity in IDS management is reduced due to the clustered tree topology which avoids overheads for loop checks and polling routines among neighboring nodes as in [34,35].

5 Conclusion and Future Work

In this paper we propose a new approach to anomaly detection and alarm generation logic using weak process models. Weak models are a simplified version of hidden markov models.

The proposed anomaly detection logic for threat modeling, threat identification, and alarm generation has been validated using MATLAB simulations.

Other threats are currently being modeled using the proposed formalism. The objective is the experimentation on a MicaZ cluster-based sensor network with a stepped implementation and deployment approach starting from few sensor nodes. Currently we are carrying on early experimentations on few MicaZ sensor nodes. Moreover we are packing ADL code for WSN development environment.

We are currently working on the capability of detecting a threat attacking nodes that are not close to each other. We showed in section 3 how the system performs when a wormhole was applied to distant nodes. We are currently extending this analysis to other attacks. Now the system uses only control messages to detect attacks. We plan to extend the proposed intrusion detection systems to include messages that contain monitoring data.

References

1. Roosta, T., Shieh, S., Sastry, S.: Taxonomy of security attacks in sensor networks. In: First IEEE International Conference on System Integration and Reliability Improvements, Hanoi, Vietnam, vol. 1, pp. 529–536 (2006)
2. Debar, H., Dacier, M., Wespi, A.: Towards a Taxonomy of Intrusion-Detection Systems. Computer Networks: The International Journal of Computer and Telecommunications Networking 31(9), 805–822 (1999)
3. Forney, G.: The Viterbi Algorithm. Proc. IEEE 61(3), 268–278 (1973)
4. Pugliese, M., Giani, A., Santucci, F.: A Weak Process Approach to Anomaly Detection in Wireless Sensor Networks. In: First International Workshop on Sensor Networks (SN 2008), Virgin Islands (2008)
5. Ephraim, Y., Merhav, N.: Hidden Markov Processes. IEEE Trans. Informormation Theory 48(6) (2002)
6. Loo, C., Ng, M., Leckie, C., Palaniswami, M.: Intrusion Detection for Routing Attacks in Sensor Networks. International Journal of Distributed Sensor Networks (2005)
7. Doumit, S., Agrawal, D.: Self Organized Critically and Stochastic Learning Based Intrusion Detection System for Wireless Sensor Networks. In: Military Communications Conference, MILCOM (2003)
8. Jiang, G.: Robust process detection using nonparametric weak models. International Journal of Intelligent Control and Systems 10 (2005)
9. Yin, Q., Shen, L., Zhang, R., Li, X., Wang, H.: Intrusion Detection Based on Hidden Markov Model. In: International Conference on Machine Learning and Cybernetics, vol. 5, pp. 3115–3118 (2003)
10. Khanna, R., Liu, H.: System Approach to Intrusion Detection Using Hidden Markov Model. In: Proceedings of the international conference on Wireless communications and mobile computing, vol. 5, pp. 349–354 (2006)
11. Sheng, Y., Cybenko, G.: Distance Measures for Nonparametric Weak Process Models. In: IEEE International Conference on Systems, Man and Cybernetics, vol. 1, pp. 722–727 (2005)
12. Giani, A.: Detection of Attacks on Cognitive Channels. Ph.D. Thesis, Dartmouth College, Hanover, NH, USA (2006)

13. Silva, A., Martins, M., Rocha, B., Loureiro, A., Ruiz, L., Wong, H.: Decentralized Intrusion Detection in wireless sensor networks. In: Proceedings of the 1st ACM International Workshop on Quality of service and security in wireless and mobile Networks (2005)
14. Karlof, C., Wagner, D.: Secure routing in wireless sensor networks: Attacks and countermeasures. In: 1st IEEE International Workshop on Sensor Network Protocols and Applications, vol. 10 (2003)
15. Rabiner, L., Juang, B.: An Introduction to Hidden Markov Models. IEEE ASSP Magazine, 4–16 (1986)
16. http://www.tinyos.net/tinyos2.x/doc
17. Pugliese, M., Santucci, F.: Pair-wise Network Topology Authenticated Hybrid Cryptographic Keys for Wireless Sensor Networks using Vector Algebra. In: 4th IEEE International Workshop on Wireless Sensor Networks Security (WSNS 2008), Atlanta (2008)
18. Whitman, M., Mattord, H.: Principles of Information Security, 3rd edn. Thomson (2009)
19. Ross, A.: Security Engineering. Wiley, New York (2001)
20. Baker, Z., Prasanna, V.: Computationally-efficient engine for flexible intrusion detection (2005)
21. Dainotti, A., Pescape, A., Rossi, P., Palmieri, F., Ventre, G.: Internet Traffic modeling by means of Hidden Markov Models. Computer Networks 54, 2645–2662 (2008)
22. Al-Subaie, M., Zulkernine, M.: Efficacy of Hidden Markov Models Over Neural Networks in Anomaly Intrusion Detection. In: Proceedings of the 30th Annual International Computer Software and Applications Conference (COMPSAC), vol. 1, pp. 325–332 (2006)
23. Wang, W., Guan, X., Zhang, X.: Modeling program behaviors by hidden Markov models for intrusion detection. In: Proceedings of 2004 International Conference on Machine Learning and Cybernetics, vol. 5, pp. 2830–2835 (2004)
24. Luk, M., Mezzour, G., Perrig, A., Gligor, V.: MiniSec: A Secure Sensor Network Communication Architecture. In: Proceedings of the Sixth International Conference on Information Processing in Sensor Networks (IPSN) (April 2007)
25. Hu, Y., Perrig, A., Johnson, D.: Packet Leashes: A Defense against Wormhole Attacks in Wireless Ad Hoc Networks. In: Proceedings of the INFOCOM 2003 (2003)
26. Law, Y., Havinga, P., Johnson, D.: How to Secure a Wireless Sensor Network. In: Proc. of the International Conference on Intelligent Sensors, Sensor Networks and Information Processing Conference (2005)
27. Ngai, E., Liu, J., Lyu, M.: On the Intruder Detection for Sinkhole Attack in Wireless Sensor Networks. In: Proc. of the IEEE International Conference on Communications, ICC 2006 (2006)
28. Krontiris, I., Dimitriou, T., Giannetsos, T., Mpasoukos, M.: Intrusion Detection of Sinkhole Attacks in Wireless Sensor Networks. In: Proc. 3rd International Workshop on Algorithmic Aspects of Wireless Sensor Networks, AlgoSensors 2007 (2007)
29. Dallas, D., Leckie, C., Ramamohanarao, K.: Hop-Count Monitoring: Detecting Sinkhole Attacks in Wireless Sensor Networks. In: Proc. of the 15th IEEE International Conference on Networks, ICON 2007 (2007)
30. Brust, M.R., Andronache, A., Rothkugel, S., Benenson, Z.: Topology-based Clusterhead Candidate Selection in Wireless Ad-hoc and Sensor Networks. In: 2nd International Conference on Communication Systems Software and Middleware, COMSWARE 2007, pp. 1–8 (2007)

31. Fok, C.-L., Roman, G.C., Lu, C.: Agilla: A Mobile Agent Middleware for Sensor Networks. Tech. Report, Washington University in St. Louis, WUCSE-2006-16 (2006)
32. Balasubramainyan, J., Garcia-Fernandez, J.O., Isacoff, D., Spafford, E., Zamboni, D.: An architecture of Intrusion Detection using Autonomous Agents, Department of Computer Science, Purdue University TR 98-05 (1998)
33. Vahid Dastjerdi, A., Abu Bakar, K.: A Novel Hybrid Mobile Agent Based Distributed Intrusion Detection System. Proc. of World Academy of Science Engineering and Technology 35 (November 2008)
34. Ramachandran, G., Hart, D.: A P2P Intrusion Detection System based on Mobile Agents. ACM, New York (2004)
35. Zhou, C.V., Karunasekera, S., Leckie, C.: A Peer-to-Peer Collaborative Intrusion Detection System. In: Proc. of 13th IEEE International Conference on Communications, ICC 2005 (2005)