Agent-Based Immunological Intrusion Detection System for Mobile Ad-Hoc Networks

Aleksander Byrski[1] and Marco Carvalho[2]

[1] AGH University of Science and Technology, Kraków, Poland
olekb@agh.edu.pl
[2] Institute for Human and Machine Cognition, Pensacola, U.S.A.
mcarvalho@ihmc.us

Abstract. Mobile Ad-hoc Networks are known to bring very special challenges to intrusion detection systems, mostly because of their dynamic nature and communication characteristics. In the last few years, several research efforts have proposed the use of immune-inspired systems for intrusion detection in MANETs. In most cases, however, only low-level pattern construction and matching have been considered, often customized to specific routing strategies or protocols. In this paper we present a more general, agent-based approach to the problem. Our approach proposes the use of artificial immune systems for anomaly detection in a way that is independent of specific routing protocols and services. After introducing the problem and the proposed system, we describe our proof-of-concept implementation and our preliminary experimental results over NS-2 simulations.

1 Introduction

Anomaly and Intrusion Detection Systems (IDS) have long been proposed in support of security strategies for computer networks. Most commonly applied in the context of enterprise networks, conventional IDS generally relies on a number of detection elements (sensors) and some (often centralized) components that correlate information among sensors to identify anomalies. Such components are responsible for learning how to identify and differentiate normal (self) patterns, from abnormal (non-self) traffic or system patterns.

Mobile Ad-hoc Networks (MANETs) are characterized by their lack of a fixed support infrastructure and their transient nature. Together, these characteristics lead to a very challenging environment for IDS implementation. Frequent changes in topology and communication patterns in MANETs require the use of specialized protocols and strategies for routing, transport and security. In particular, the use of autonomous agents performing the duties of a single security detector and being able to communicate with neighboring agents to share information and inferences is well suited for IDS implementation in MANETs.

Biologically-inspired approaches for anomaly detection systems have proven to be very interesting, often yielding very effective results [1] for some applications.

M. Bubak et al. (Eds.): ICCS 2008, Part III, LNCS 5103, pp. 584–593, 2008.

In particular, immune systems-based detection and defense mechanisms seem to provide a good analogy to the requirements and capabilities expected from an IDS for these kinds of environments. The approach proposed in [2], however, like most others, is defined for a specific routing protocol. In the paper we propose a more general approach to the problem. We first provide a brief introduction on the state-of-the-art in Intrusion Detection Systems for MANETs. After presenting the architecture and core components of the proposed systems, we introduce and discuss the modelling of behavior patterns. We conclude the work with by presenting and discussing our preliminary experimental results and our conclusions.

2 Intrusion Detection System for MANET

The primary goal of an Intrusion Detection Systems (IDS) is to detect the unauthorised use, misuse and abuse of computer systems and networks resources. The earlier research on IDS dates back from the 80's [3], when it basically aimed on providing auditing and surveillance capabilities to computer networks. Following that idea, the first generic intrusion detection model [4] was proposed in 1987.

The basic implementation of that model consisted in a real-time expert system whose knowledge was derived from statistical inference based on the audit trails of users or system resources. It stored characteristics describing the normal behavior of subjects with respect to objects and provided the signature of abnormal behaviors - a statistical metric and model were used to present profiles. As a subject, an individual system user, a group of system users or the system itself can be considered, while objects can be files, programs, messages, records, terminals etc. When a subject acts upon a specific object, it usually generates an event, which alters the statistical metric state of both subject and object. A knowledge base contains activity rules to be fired for updating profiles, detecting abnormal behaviour, and producing reports. An inference engine works by triggering rules matching profile characteristics. Since then, various IDS have been developed and a number of intrusion detection systems have directly employ this model e.g. [5] [6].

Mobile ad-hoc networks (MANET) are self organized networks without any predefined structure (other that the end users are equipped with radio-based networking interfaces). Communication beyond the transmission range is made possible by having all the nodes serve as routers. They should participate in common routing protocol (such as AODV). This makes these networks very difficult to perform monitoring, because of dynamical reorganization of the topology. In classical (non ad-hoc) networks possible reasons for node misbehavior may be caused by faulty software or hardware, sometimes caused by a human intruder. Other treats arise for ad-hoc networks, i.e. misuse of the routing protocols [7] [8].

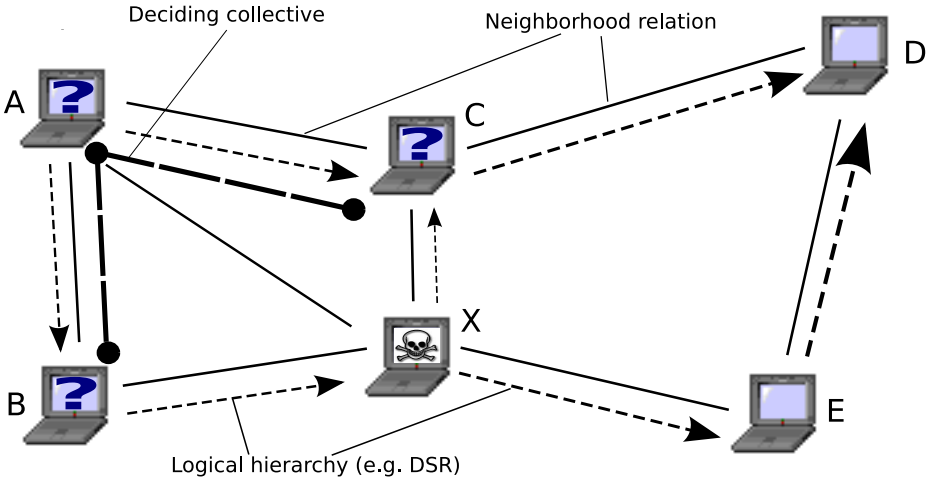


Fig. 1. Structure of distributed IDS

MANETs seem to pose special requirements for IDS, because of [7]:

- Mobility and dynamism – nodes in MANET are highly mobile and topology changes in sometimes unpredictable manner.
- Lack of fixed traffic points – there are no firewalls or routers as in classical computer networks, all nodes are used as routers.
- Limitations of host-resident network intrusion detection – detectors may also become the target of an attack per se, or by distracting of their communication protocol.
- Wireless communication – RF medium is susceptible to eavesdropping, jamming, interference and many other MAC threats what may effect in loss of packets and intermittent connectivity.
- Resource constraints – the resources vital co communicating in MANET environment are limited, e.g. energy (battery operated nodes), varying throughput because of dynamic topology configuration.

So the IDS for MANETs must be decentralized, with some level of data aggregation and information sharing – e.g. the detectors may consult themselves in order to evaluate the accuracy of detection and provide better responses.

Sterne e.a. [7] propose a reasonable solution to the problem based on the hierarchical organization of detectors. The dynamic nature of MANETs, however, tend to complicate the creation of (virtual) dynamic structure, often compromising this kinds of approaches. A far more simple and yet effective approach for detecting unfavourable behaviors might be considered by using non-hierarchical approaches similar to ethically-social mechanisms of decision undertaking, proposed in [9].

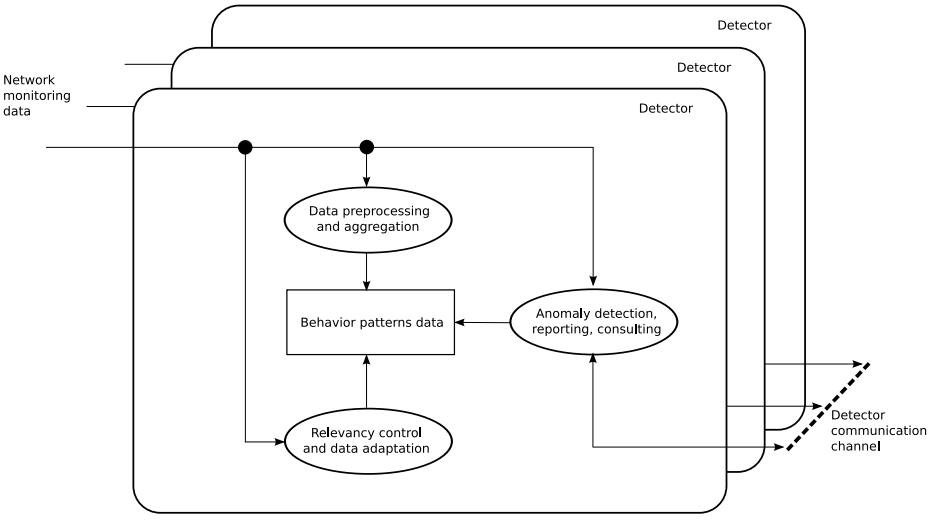


Fig. 2. Detection algorithm

3 Agent-Based IDS for MANET

3.1 System Structure Overview

Proposed IDS consists of a set of detectors (that may be perceived as intelligent agents, because of their autonomy [10]) introduced into the system (i.e. several nodes that take part in normal routing of the packets are considered as detectors). After sensing some kind of disturbance in the behavior of certain nodes, the detectors should try to reach neighboring detectors and communicate with them, in order to consult their observation. Then the decision of raising an alarm may be undertaken.

In this way constructed system allows to preserve no hierarchy and to undertake the decision based on asking several (possible) neighboring detectors for an opinion (see Fig. 1) what ensures reliability of the approach (even if the detectors cannot maintain contact among themselves, they still may react to the behavior they sense).

The behavior of the intruder node X is evaluated by its neighbors. Based on the overhearing of the X's transmission, the decision algorithms implemented inside nodes A, B and C, after consulting among them any of these nodes may report the invader to specific authorities (what will usually involve sending a message to the administrator).

3.2 Detection Algorithm Overview

The main task of the detectors is to perform monitoring of the routed and overheard packets (that are received by neighboring nodes) and build a certain model

of normal (or abnormal, depending on the actual detection algorithm used) behavior of the system. Then, current behavior of certain nodes is evaluated based on the model. When a sudden change of certain node's behavior occurs, the alarm or need for consulting is raised.

Specific algorithm of adapting the collected normal behavior should be also considered. Generally the course of the algorithm should be optimized in order to sense fast changes in the behavior of neighboring nodes, and to adapt to the slow ones.

In Fig. 2 the structure of the detector is shown. General aspects of the algorithm must be supplemented with specific anomaly detection algorithm that would be able to construct the behavior model and to perform certain reasoning in order to classify unknown behavior of the neighbors.

4 Behavior Model and Anomaly Detection

4.1 Behavior Pattern Model

One of the most important thing in IDS is to propose specific behavior pattern creation what would let to evaluate neighbor behavior. Le Boudec and Sarafijanovic propose the approach based on classification of aggregated count of packets overheard during specific period of time [2].

The approach however considers only one routing protocol (AODV), what makes their approach improper for the other popular protocols (e.g. OLSR or ZRP). Le Boudec and Sarafijanovic use extensively biological inspiration, though the universal approach should be independent of the detection algorithm. Besides, following algorithm should allow to consider any routing protocol in order to create more adaptable and universal IDS. In this section, the algorithm for constructing behavior patterns for the nodes in MANET will be presented.

In order to capture the behavior in a certain period of time, first, specific packet signature is constructed. Packet signature is a way of describing certain number of similar packets overheard in the network. Packet signature may be described as a vector of values

$$PS = ATR^k \tag{1}$$

where k is length of the packet signature and ATR is one of the spaces described below (in fact the contents of this Cartesian product may be further adapted and extended according to the specific type of network):

- SRC, DST – source and destination identification, may be IP address, MAC address or other unique ID ($SRC, DST \subseteq \mathbb{N}$).
- DIM – distance mark describing how far (e.g. in hops) are interlocutors (when the protocol allows to get this information) ($DIM \subseteq \mathbb{N}$).
- PTF, PTT – port number from (to) describing the range of the ports that the packet is sent from ($PTF, PTT \subseteq \mathbb{N}$).
- PYS – payload size ($PYS \subseteq \mathbb{N}$)).

– PYT – payload type ($PYT \subseteq \mathbb{N}$)).

E.g. packet signature may look as follows:

$$PS_1 = (atr_1, atr_2, \ldots, atr_7) = (10, 12, 5, 1003, 1005, 128, 'CBR') \tag{2}$$

being a vector described in the following packet signature space:

$$PS = SRC \times DST \times DIM \times PTF \times PTT \times PYS \times PYT \tag{3}$$

In order to capture the behavior during specific time, packet signatures are aggregated based on the receiver's ID and presented in the following form:

$$B_1 = \{(PS_1, NO_1), (PS_2, NO_2), \ldots\} \tag{4}$$

where B_i is behavior of the node i and NO_i is number of PS_i gathered in a specific period of time (it may be also frequency or value any other function dependent on the number of packet signatures). B_i is in fact a vector described in the following space:

$$B = APS^k = (PS \times \mathbb{R})^k \tag{5}$$

where:

– k is maximal number of packet signatures aggregated in one behavior pattern.
– APS is aggregated packet signature (value describing number of packet signatures is added at the end of the vector).

Packets which are aggregated into a specific group being the part of the behavior based on certain similarity measure:

$$SIMAPS : APS^2 \to \mathbb{R} \tag{6}$$

Range of this function may be constrained (e.g. to the interval $[0, 1]$) in order to clearly state the maximal, minimal and medium values of similarity. This similarity function depends on the following similarity measure used to discover whether two attributes are similar:

$$SIMATR : ATR^2 \to \mathbb{R} \tag{7}$$

In order to evaluate the similarity of the behavior patterns (what is needed to implement several detector algorithms, e.g. immunological–based ones) similar function should be defined:

$$SIMB : B \times B \to \mathbb{R} \tag{8}$$

Range of this function may also be constrained (e.g. to the interval $[0, 1]$) for the same reason as mentioned above.

4.2 Anomaly Detection Algorithm

Although any anomaly detection algorithm may be employed by detector, for the current prototype implementation and generation of experimental results, immune-based anomaly detection algorithm was used. Based on the several similar approaches presented i.a. by [1] negative selection algorithm was used.

Negative selection requires construction of self and non-self behavior patterns. Self patterns are constructed in a way described in 4.1. Every detector maintains a dataset with the collection of self patterns (normal behavior) collected during normal course of network operation, and non-self patterns (anomalous behavior) which are generated randomly with use of specific similarity measure. I.e. the non-self set of behavior patterns contains only these patterns that are not similar to any of self patterns (by the means of similarity function described by equation 8). One of possible implementation of this similarity function may look as follows:

$$SIMB(bp_1, bp_2) = \frac{\sum_{aps_1 \in bp_1, aps_2 \in bp_2} SIMPS(aps_1, aps_2)}{\#bp_1 \cdot \#bp_2} \tag{9}$$

where:

- $bp_1, bp_2 \in B$
- $\#bp_1$ is count of elements in the set bp_1 (count of aggregated packet signatures).

Using this equation the similarity of the two behavior patterns may be determined. The denominator was introduced in order to scale the output to the interval $[0, 1]$, so, for the same patterns the function will return value 1. In order to complete the definition, $SIMPS$ function must be stated, e.g. as follows:

$$SIMAPS(aps_1, aps_2) = \frac{1}{k+1} \cdot \#S \tag{10}$$

where:

- $S = \{(atr_{1i}, atr_{2i}) | SIMATR(atr_{1i}, atr_{2j}) > t\}$ – is a set of tuples containing corresponding attributes of aps_1 and aps_2 (the same value of index i),
- $t \in [0, 1]$ is a similarity threshold,
- $i \in \mathbb{N}$.

After collecting of the self behavior patterns the detector starts to monitor the communication of the neighboring nodes and report the anomalous behavior (behavior that is similar to one of its non-self pattern) to neighboring detectors (consulting) or to the end-user (alarm).

During the consulting some of the non-self patterns may be exchanged among the detectors, in order to spread the knowledge about behavior throughout the detectors set.

4.3 Collective Decision

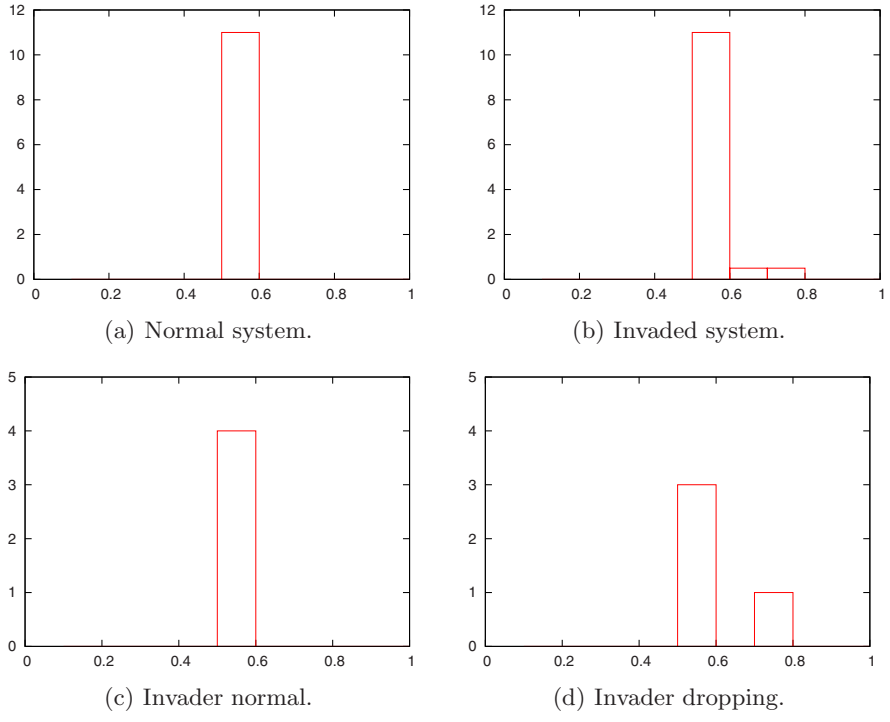
After stating that the behavior of the observed neighbor is unfavourable, the procedure of collective decision is started, that consists in consulting of the neighboring detectors, when detector discovers that the observed behavior of a neighbor is anomalous. When the answer to the question is returned, the detector includes it into consideration. Then the possible action of the detector may be determined using different collective intelligence managing techniques (e.g. Winner Takes All, when the decision of the most reliable neighbor is the most important in consultation, or Winner Takes Most where the average decision of the neighbors is taken into consideration) inspired by [9].

The decision undertaken by a detector should however base not only on the collective voting techniques because in the environment where no-one has found any intruders, the collective will never decide to raise an alarm when one of the members will find an intruder. Another thing is the autonomy of the agents-detectors, which relieves them from relying completely on external information. Instead, the detector should maintain a database describing the behavior of its observed neighbors. It must be of course dynamically modified because of the changes of the network topology. The information contained there will be volatile. Anyway, after spotting several subsequent unfavorable activities of the neighbor, the detector should raise an alarm without consulting the collective.

Voting-based techniques require some sort of global control mechanism (that should assign the weights to the detectors), which is undesirable in this kind of distributed environment. The one rational possibility is to introduce second level of detectors, so called ,,super-detectors", that should maintain a database of their neighboring detectors and apply specific reliability weight that might be used in order to help the collective to undertake the decision. In this way hierarchical structure of the detectors will be introduced, however it should not be strict because of the dynamic nature of the network. Instead, the super-detector should be chosen collectively from the group of neighboring detectors and their function might not rest forever (they might be reduced in the future to the role of simple detector). The reliability weight of the certain detector might be changed only by the super-detector. These ideas are now considered as subjects of further research.

4.4 Adaptation and Pattern Exchanging

Detector maintains his own measurements of the collective agreement (e.g. in a very simple case it stores the information, how many times his own decision was similar to the decision of the collective). After observing the measure of the collective decision and changes in the input data (using specific self–pattern matching measures) it may try to send or acquire some of behavior patterns and broadcast an offer to perform such an exchange to neighboring detectors. The detector may also decide to drop some of its patterns and regenerate them from scratch in order to adapt to the changes present in the environment.



(a) Normal system. (b) Invaded system.

(c) Invader normal. (d) Invader dropping.

Fig. 3. Count of the test patterns highly similar to non-self patterns (Y-axis) in the similarity range (X-axis) for the whole system (a,b) and invader (c,d)

5 Preliminary Experimental Results

The simulation was performed with using NS-2 network simulator. MANET routing protocol AODV was used along with 802.11 wireless communication. Specific simulation environment consisted of 30 agents organized in three concentric circles, rotating in different directions. There was one node (detector) in the center that received the information sent from one node located outside the circles. Normal behavior of the environment consisted in observing the transmission by the detector, building a behavior model during 100 s of simulation. In order to simulate anomalous behavior, one node from the most central circle stopped forwarding (started dropping) the packets after 50 s of simulation. The behavior pattern results were collected and displayed in tables and histograms presented below. Histograms presented In Fig. 3 show the number of the non-self matching of the behavior patterns collected in the system with normal and anomalous behavior. The data was collected for two observed nodes. The examined test results were gathered for the whole system before (see Fig. 3(a)) and after intrusion (see Fig. 3(b)). The graph changes, there are more patterns similar to non-self patterns during the intrusion. Then the evaluation of single

intruder node was performed. Comparing Fig. 3(c) and Fig. 3(d) yields, that higher matching among test patterns and non-self patterns occurs during the intrusion.

6 Conclusion

In this paper we have introduced and discussed an agent-based architecture of IDS for MANETs. In our approach, an intelligent agent-based system is augmented with an immune system-based anomaly detection algorithm. Our preliminary NS-2 based experimental results were encouraging, and seem to indicated that the proposed system can be effectively used to detect abnormal behavior in MANET environments. In continuation of this effort, we will expand our simulation analysis to include more complex network scenario and traffic patterns.

References

1. Dasgupta, D.: Artificial Immune Systems and Their Applications. Springer, New York (1998)
2. Boudec, J.Y.L., Sarafijanovic, S.: An artificial immune system approach to misbehavior detection in mobile ad hoc networks. In: Ijspeert, A.J., Murata, M., Wakamiya, N. (eds.) BioADIT 2004. LNCS, vol. 3141, pp. 396–411. Springer, Heidelberg (2004)
3. Anderson, J.: Computer security threat monitoring and surveillance. Technical report, James P. Anderson Co., Fort Washington, PA (1980)
4. Denning, D.E.: An intrusion-detection model. IEEE Trans. Softw. Eng. 13(2), 222–232 (1987)
5. Ilgun, K., Kemmerer, R., Porras, P.: State transition analysis: A rule-based intrusion detection approach. Software Engineering 21(3), 181–199 (1995)
6. Jackson, K., DuBois, D., Stallings, C.: An expert system application for detecting network intrusion detection. In: Procedings of the 14th National Computer Security Conference, pp. 215–225 (1991)
7. Sterne, D., et al.: A general cooperative intrusion detection architecture for manets. In: IWIA 2005: Proc. of the Third IEEE Int. Workshop on Information Assurance (IWIA 2005), pp. 57–70. IEEE Computer Society, Los Alamitos (2005)
8. Drozda, M., Szczerbicka, H.: Artificial immune systems: Survey and applications in ad hoc wireless networks. In: Proc. of the 2006 Int. Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2006), Calgary, Canada, pp. 485–492 (2006)
9. Rojek, G., Cieciwa, R., Cetnarowicz, K.: Algorithm of behavior evaluation in multiagent system. In: Sunderam, V.S., van Albada, G.D., Sloot, P.M.A., Dongarra, J. (eds.) ICCS 2005. LNCS, vol. 3516, pp. 711–718. Springer, Heidelberg (2005)
10. Bradshaw, J.M. (ed.): Software Agents. AAAI Press/The MIT Press (1997)