One Size Does Not Fit All – How to Approach Intrusion Detection in Wireless Sensor Networks

Andriy Stetsko and Václav Matyáš

Department of Computer Systems and Communications
Faculty of Informatics, Masaryk University
{xstetsko, matyas}@fi.muni.cz

Abstract. A wireless sensor network (WSN) is a highly distributed network of resource constrained and wireless devices called sensor nodes. In the work we consider intrusion detection systems as they are proper mechanisms to defend internal attacks on WSNs. A wide diversity of WSN applications on one side and limited resources on other side implies that "one-fit-all" intrusion detection system is not optimal. We present a conceptual proposal for a suite of tools that enable an automatic design of intrusion detection system that will be (near) optimal for a given network topology, capabilities of sensor nodes and anticipated attacks.

1 Introduction

A wireless sensor network (WSN) consists of sensor nodes – devices that are equipped with sensor(s), microcontroller, wireless transceiver and battery. Each sensor node monitors some physical phenomenons (e.g., humidity, temperature, pressure, light, etc.) inside an area of deployment. The collected measurements are then sent to a base station – a gateway between a WSN and external world (in most cases the Internet).

In the work we consider WSNs that contain hundreds of thousands of nodes distributed over an area of hundreds square kilometers. Communication range of sensor nodes is limited to tens of meters and hence not all of them can directly communicate with a base station. Therefore, data are sent hop-by-hop from one sensor node to another until they reach a base station (see Figure 1).

Sensor nodes are constrained in processing power and energy, whereas a base station is assumed to have laptop capabilities and unlimited energy resources. Crossbow MICAz[1] is an example of average sensor node. It contains Atmel Atmega128L microcontroller, 802.15.4 compliant (250kbps) Texas Instruments CC2420 transceiver and two AA batteries. The microcontroller features 8b processor (operating at 8MHz), 128kB FLASH, 4kB EEPROM and 4kB SRAM. Currently the sensor node is available at price of €110. That eliminates deployment of a large number of sensor nodes. However, it is believed that recent advances in micro-electro-mechanical systems will decrease the cost significantly.

[1] See manufacturer's website http://www.xbow.com/.

Mathematical and Engineering Methods in Computer Science (MEMICS), Znojmo, Czech Republic, 2009.
Petr Hliněný, Vashek Matyáš, Tomáš Vojnar (Eds.)
Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Germany
Digital Object Identifier: 10.4230/DROPS.MEMICS.2009.2347

It is expected that WSNs will have many applications in military, ecology, building and industrial automation, energy management, agriculture and even wildlife monitoring. Security becomes an important issue for WSNs and brings new challenges for security engineers.

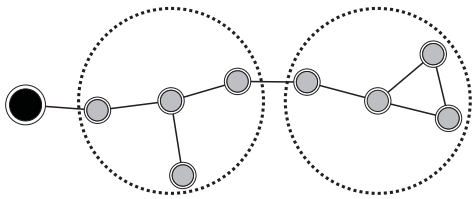


Fig. 1: Wireless sensor network. A base station is depicted as the black filled circle and sensor nodes are depicted as gray ones. We assume that communication ranges (represented by dotted circles) of neighboring sensor nodes are symmetric.

Cryptographic techniques can be used to prevent an external attacker (outsider) [9] from eavesdropping or altering the ongoing communication[2]. Encryption does not solve the problem of jamming attacks, where a malicious node (or other device) purposefully tries to interfere with physical transmission and reception of wireless communication.

An area of deployment is most often not physically protected and an attacker can easily access the area and capture some nodes[3]. Being a legitimate participant of the network the attacker (insider) can launch a variety of internal attacks. In the work we consider: a selective forwarding attack in which an attacker selectively drops packets [5]; a sinkhole attack in which an attacker attracts all traffic from a particular area towards itself, typically by making a compromised node look attractive to neighboring nodes with respect to routing algorithm [5]; a packet alternation attack in which a malicious node modifies packets that it forwards for the neighbors.

Sensor nodes are not tamper-resistant and an attacker can extract cryptographic keys from captured nodes. The attacker can replicate (also known as clone attack) [6] the nodes, deploy them into a network and then launch attacks described above. The attacker can also create nodes with several identities, also known as Sybil nodes [5]. These nodes may have an impact on multipath routing, voting, data aggregation, fair-resource allocation and misbehavior detection.

In this work we consider intrusion detection systems (IDSs) since they are, in comparison to cryptographic techniques, better mechanisms to defend against internal attacks on WSNs. In Section 2 we describe basics of intrusion detection systems for wireless sensor networks – what kinds of audit data can be gathered

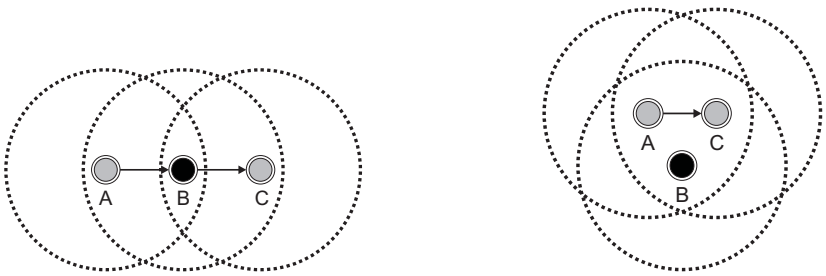
[2] A survey on performance of symmetric/asymmetric cryptographic primitives and hash functions implemented for WSNs is available in [8].

[3] We assume that a number of such nodes is significantly smaller than a total number of sensor nodes in the network.

and for detection of what types of attack they can be used. "One-fit-all" IDS is not optimal because of the wide range of WSN applications and limited resources of sensor nodes. In Section 3 we propose a conceptual architecture of a suite of tools that will provide administrators with an IDS that fits best its purposes.

2 Intrusion detection in wireless sensor networks

In the work we consider a distributed IDS that consists of IDS agents. We assume that every sensor node runs an IDS agent which monitors its neighbors using both local and watchdog monitoring techniques [1]. In the local monitoring technique sensor nodes collect and analyze only data forwarded by themselves (see Figure 2a). In the watchdog technique, sensor nodes collect an analyze data overheard in their neighborhood (see Figure 2b). We assume that sensor nodes employ single-channel transceivers. However, if the multi-channel transceivers are used, it might happen (the worst scenario) that the watchdog technique will be useless and an IDS will have to rely only on the local monitoring technique.



(a) The sensor node B monitors traffic that it forwards from the node A to the node C

(b) The sensor node B monitors in promiscuous mode traffic from the node A to the node C

Fig. 2: Traffic monitoring techniques

A conceptual model of an IDS agent is presented in [10]. Audit data gathered by a local audit data collection module are subsequently analyzed by a local detection module(s). A cooperative detection module is used to propagate intrusion detection state information or/and audit data among neighboring nodes. In case a local detection evidence is weak or inconclusive the cooperative detection module can use information (e.g., audit data) received from other IDS agents to detect an ongoing attack. After the attack is detected local response and global response modules trigger reactions. We assume that the local response module will stop any communication with the malicious node. The global response module will notify an administrator and he/she will remove or reprogram the malicious node. A secure communication module should provide cooperating nodes with a secure communication channel.

We assume that IDS agents will be implemented for TinyOS – the most widely used operating system for wireless sensor networks. It is important to understand what kind of network audit information we can gather. In TinyOS, the basic network abstraction is an active message that includes source and destination addresses [3]. Also it provides synchronous acknowledgements. Hardware independent components (e.g., active message) are built on top of hardware dependent components. Crossbow Imote2, MICAz and TELOSB sensor nodes as well as Sentilla Tmote Sky sensor nodes use TI CC2420 transceiver[4]. The corresponding CC2420 Radio Stack [4] supplies each outgoing message with a unique data sequence number and also provides possibility to read RSSI (received signal strength indication) of each received packet.

The audit data collection module logs source and destination addresses of received, overheard and sent packets. Also it logs: RSSI of received packets as well as information whether a packet passed CRC check or not; information whether an attempt to send a packet was successful or not, whether a packet was received by recipient (acknowledged) and how much time was spent waiting for the channel (carrier sensing time). All this information is gathered for a period of time of duration T and we call it temporal information. Having this information, different temporal statistics can be calculated. Examples of such statistics are presented below.

1. Packet delivery ratio – a ratio of packets that are successfully delivered to a destination compared to a number of packets that were sent by the sender.
2. Packet sending rate – a number of packets sent by a neighboring sensor node.
3. Packet receiving rate – a number of packets received by a neighboring sensor node.
4. Packet dropping rate – a ratio of packets sent by a neighboring sensor node with respect to a number of packets received by that node.
5. Number of neighbors.
6. A function of RSSI (e.g., average, maximum).
7. A function of carrier sensing time (e.g., average, maximum).

Packet delivery ratio can be calculated either at the sender or at the receiver. At the receiver it is calculated as a ratio of a number of packets that passed the CRC check with respect to a number of packets received. At the sender it is calculated as a ratio of a number of ACK received with respect to a number of packets sent.

The calculated statistics can be used to detect different attacks. For example, carrier sensing time, packet delivery ratio and RSSI are used to detect jamming attacks [2]. Selective forwarding attacks can be detected by monitoring packet dropping rate [1]. Packet receiving ratio can be used to detect sinkhole attacks.

[4] We consider these radio chips since they are 802.11.4 compliant and provide hardware support of AES.

3 Intrusion detection system fits to purpose

Density of a network, capabilities of involved sensor nodes, anticipated types of attack and other critical parameters may vary from one application to another. Due to a wide range of WSN applications on one side and limited resources on other side a "one-fit-all" IDS is not optimal. Therefore, we propose to make a two-level optimization.

1. In order to detect different attacks a variety of local detection modules can be implemented. An administrator will specify anticipated attacks and we propose to include only such detection modules in the IDS agent that detect the specified attacks. This approach will save memory and energy that are very important for WSNs. To our best knowledge nobody has yet applied such idea to WSNs.

2. Parameters of detection modules (included in the IDS agent configuration) might not be optimal for a given application. We propose to optimize them for a network topology, sensor nodes capabilities and anticipated types of attack, which all will be specified by a network administrator. In conventional networks, in majority of cases, a trade-off between a number of false positives, a number of false negatives and memory usage is found. However, for WSNs this is not enough. Sensor nodes are energy constrained and if ever depleted they will stop fulfilling their main goal – monitoring of area of deployment. Therefore, for WSNs a trade-off between detection accuracy, memory usage and energy usage should be found. For example, if IDS agents cooperate between themselves it involves communication, which in comparison with computation, consumes significantly more energy [7]. On other side cooperation increases a detection accuracy since a single monitoring node may not have enough information to detect an attack, e.g., due to collisions [1].

We propose a suite of tools that should provide an administrator of a network with a (near) optimal IDS. The suite includes `Framework` and `Simulator` (see Figure 3). A network administrator provides descriptions of network topology, sensor node characteristics and anticipated attacks to the `Framework`. It contains a database of available components which will be used to compose an IDS agent. To make automatic design of an IDS agent possible we should specify types of component that can be used and interfaces between them. The first step is to design a local audit collection module that will gather audit data that might be "ever" required by any detection module. We have undertaken the analysis of state-of-the-art IDSs for WSNs and possible audit data ever met in the studied literature have been described in Section 2. In the worst case, if some detection module needs audit data that are not gathered by the local audit collection module, the collection module should be designed in such way that an administrator will be able to add the required functionality easily.

There are different detection modules among the components in the database. Based on the specified attacks the `Framework` generates a possible configuration of the IDS agent. The configuration will be optimized using the `Simulator` in the following way. The `Framework` sets initial values of parameters and evaluates effectiveness of the configuration using metrics, examples of which we describe

further in the section. Based on the evaluation the `Framework` "improves" the values of parameters and repeats the procedure until they become (near) optimal for given network topology, capabilities of sensor nodes and anticipated attacks. Should there are more than one possible configuration each of them is optimized separately. Evaluations of optimized configurations can be used by an administrator to choose the one that fits best its purposes.

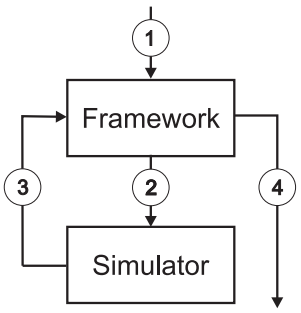


Fig. 3: An architecture of the proposed suite of tools. The arrow with number "1" depicts inputs provided by an administrator. The arrow "2" depicts a configuration that is passed to the `Simulator`. The arrow "3" depicts a feedback on effectiveness of the configuration based on evaluating metrics. The arrow "4" depicts an optimized configuration and its evaluation of effectiveness.

In order to understand how to evaluate an IDS we will firstly analyze attacks and their impacts on a network. Sinkhole attacks result in data receiving delays and additional energy usage because data do not travel along the shortest path. Let us assume that each node shares a cryptographic key with a base station as well as each node sends packet and waits for its acknowledgement. If a packet is modified along the path a base station may drop the packet and request to resend it again. That will cause a packet delay and additional usage of energy. If the packet is modified again and again it will not be ever delivered to the base station. Selective forwarding attack may result in sending the dropped packet again and again. Similarly as in the packet alternation attack, that can cause delays and additional energy usage. If a jamming attack lasts too long a packet will be dropped if a new packet arrives and buffer is full. *To sum up, the considered attacks may cause losses of packets, modifications of packets, delivery delays and additional energy usage.*

The main goal of IDS is to detect ongoing attacks and respond in such a way that the impact of the attacks will be minimal. The presented examples of metrics evaluate the impact of the sinkhole, selective forwarding, packet alternation and jamming attacks as well as an effectiveness of a given IDS. The smaller the measured impact is, the more effective IDS is. We assume that all of the metrics will be measured during a period of time of duration Q ($Q \in R^+$). The parameter Q will be chosen by an administrator.

1. *We propose to count the number of lost packets* as a difference between a number of packets sent by sensor nodes and a number of packets successfully received by a base station. The packets may get lost due to collisions, buffer overflows and environment changes. We assume that a number of packets lost in such way is constant for each time interval of duration Q if Q is long enough. It is noteworthy to mention that an increase of number of detected malicious nodes does not necessary mean that a number of successfully received packets will increase as well. Having a smaller number of nodes an attacker can change the strategy and drop/modify/jam more packets than before.

2. *We propose to count the number of modified packets received at the base station.* Packets that had been modified and hence were subsequently dropped are not counted as they are considered as lost. We can extend the metric by introducing a function that will determine how much the original packet differs from the modified one. An administrator will specify the function and thereby specify what packet fields are more critical than others.

3. *We propose to evaluate the total amount of energy used by the network* by summing up energy used by each sensor node. However, the metric does not take into account a distribution of energy usage – some sensor nodes may be depleted soon whereas others may remain fully charged. That may cause partition of a network and measurements from isolated regions of the network will never reach a base station. Such packets will be considered as lost. In order to avoid such situations we propose to use a metric that prioritizes IDSs which detect and respond to attacks in such way that energy consumption is distributed uniformly as much as possible. As an example of the metric we consider $\Theta = \sum_{i=1}^{n} c^{e_i}$, where n is a number of sensor nodes in a network, e_i is an amount of energy consumed by sensor node i, $(1 \leq i \leq n)$ and c $(c \in R, c > 1)$ is a constant that should be specified by an administrator.

4 Conclusions and further work

IDSs are useful for different networks since there are no guaranties that an attacker remains outside a perimeter secured by a firewall. In WSNs risks of being attacked by an insider are higher than in conventional networks since the area of their deployment is most often not physically protected. Due to a distributed nature of WSNs and severe limitations of sensor nodes on energy, memory and computation power, traditional IDSs are not applicable to the WSNs. Moreover, "one-fit-all" IDS is far away from being optimal for a given network topology, capabilities of sensor nodes and set of anticipated attacks in terms of detection accuracy and resources consumption. Therefore, the aim of our work was to propose a conceptual architecture of a suite of tools that will provide a network operator with a (near) optimal IDS for a given application. The optimalization process was divided into two steps. The first will save memory and energy by including into an IDS agent only modules that are used for detection of anticipated attacks. The second will find a trade-off between detection accuracy and resources consumption by setting parameters of detection modules and evalu-

ating the configuration according to the defined metrics using a simulator. The proposed metrics evaluate impact of sinkhole, selective forwarding, packet alternation and jamming attacks by counting a number of lost/modified packets and energy consumed by a network. The smaller the measured impact is, the more effective an IDS is. The list of metrics is not complete and we currently extend it as well as add a classification of types of components that can be used to construct an IDS agent and define interfaces between them. We also plan to work on the selection of a proper optimization algorithm and a proper simulator. Since the space of possible solutions might be too large for exhaustive search, approximation algorithms might be used. Evaluation of the implemented suite of tools will be based on the time needed to find (near) optimal solution and on how close the obtained solution is to the optimal one.

5 Acknowledgement

This work was supported by the project 102/09/H042 "Mathematical and Engineering Approaches to Developing Reliable and Secure Concurrent and Distributed Computer Systems" of the Czech Science Foundation.

Also we would like to thank Petr Švenda for the fruitful discussions.

References

1. Krontiris, I., Dimitriou, T., Freiling, F. C.: Towards Intrusion Detection in Wireless Sensor Networks. In: 13th EuropeanWireless Conference. (2007)
2. Xu, W., Trappe, W., Zhang, Y., Wood, T.: The feasibility of launching and detecting jamming attacks in wireless networks. In: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing. (2005) 46–57
3. Levis, P.: Packet Protocols. (http://www.tinyos.net/)
4. Levis, P.: CC2420 Radio Stack. (http://www.tinyos.net/, 2007)
5. Karlof, C., Wagner, D.: Secure routing in wireless sensor networks: Attacks and countermeasures. In: First IEEE International Workshop on Sensor Network Protocols and Applications. (2003) 113–127
6. Parno, B., Perrig, A., Gligor, V.: Distributed detection of node replication attacks in sensor networks. In: IEEE Symposium on Security and Privacy. (2005) 49–63
7. Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D., Pister, K.: System architecture directions for networked sensors. In: Proceedings of the ninth international conference on Architectural support for programming languages and operating systems. (2000) 93–104
8. Roman, R., Alcaraz, C., Lopez, J.: A survey of cryptographic primitives and implementations for hardware-constrained sensor network nodes. In: Mobile Networks and Applications Journal, Vol. 12, Num. 4. (2007) 231–244
9. Roosta, T., Pai, S., Chen, P., Sastry, S., Wicker, S.: Inherent Security of Routing Protocols in Ad-Hoc and Sensor Networks. In: IEEE Global Telecommunications Conference. (2007) 1273–1278
10. Zhang, Y., Lee, W.: Intrusion detection in wireless ad-hoc networks. In: Proceedings of the 6th annual international conference on Mobile computing and networking. (2000) 275–283