An Intrusion Detection System for Critical Information Infrastructures Using Wireless Sensor Network Technologies

Luigi Coppolino, *Member IEEE*, Salvatore D'Antonio, *member IEEE*, Luigi Romano, *member IEEE*, and Gianluigi Spagnuolo

***Abstract*-- Wireless Sensor Network (WSN) technology is being increasingly used for data collection in Critical Infrastructures (CIs). The paper presents an Intrusion Detection System (IDS), which is able to protect a CI from attacks directed to its WSN-based parts. By providing accurate and timely detection of malicious activities, the proposed IDS solution ultimately results in a dramatic improvement in terms of protection, since opportunities are given for performing proper remediation/reconfiguration actions, which counter the attack and/or allow the system to tolerate it. We present the basic ideas, discuss the main implementation issues, and perform a preliminary experimental campaign. Not only have experiments demonstrated the effectiveness of the proposed approach in protecting the system against two very serious attacks to WSNs (namely: sinkhole, and bogus packet), but they have also proved that the stringent requirements (in terms of limited availability of resources) which are typical of current state-of-the-art WSN technologies, are met.**

***Index Terms*—Intrusion Detection, Wireless Sensor Networks, Critical Infrastructures, Critical Information Infrastructures.**

I. Introduction

TRADITIONAL Critical Infrastructures (CIs) were intrinsically secure systems, due to a combination of factors, some of which are briefly described in the following:

- they consisted (almost exclusively) of special purpose devices, which were based on proprietary technologies;
- individual sub-systems operated almost in isolation, i.e. they did not interact with the external world, with the exception of the system being controlled;
- they were largely based on dedicated (as opposed to shared) communication links;
- they massively relied on proprietary (as opposed to open) communication protocols.

These trends have been largely subverted, and it will be even more so in the future. First, Wireless Sensor Networks (WSNs) have become an integral part of virtually any CI [9]. Second, Commercial-Off-The-Shelf (COTS) components are being massively used for implementing Supervisory Control And Data Acquisition (SCADA) systems [6]. Third, subsystems are being connected using the infrastructure of the corporate Local Area Network (LAN), or even Wide Area Network (WAN) links, possibly including the public Internet, as well as wireless/ satellite trunks.

The typical architecture of current CIs has a hierarchical structure, which integrates heterogeneous devices and network trunks, also via shared network connections. To achieve interoperability, open communication protocols are being increasingly used, thus exposing SCADA systems to the same vulnerabilities which threaten general purpose Information Technology (IT) systems.

Evidence is showing that current CIs are already exposed to major security risks. As an example, in [5] it is reported that Cyberspies have penetrated the U.S. electrical grid and installed malicious software programs that could be used to disrupt the system. More in general, if one gets an opportunity to talk privately to the personnel in charge of IT security at electric utility companies or at the Department of Homeland Security, they say that they are extremely worried about security exposure of their SCADA systems.

It is worth emphasizing that we are witnessing a dramatic increase of external borne security incidents, while internal are basically stable, and accidental have increased only slightly (most probably, such a slight increase is mainly due to the increased complexity of the equipment, which results in more operator mistakes and interactions faults in general) [15].

In particular, the shared communication infrastructure has become an obvious target for disrupting a SCADA network. For example, an attacker may exploit a vulnerability of the wireless trunk of a SCADA communication infrastructure to prevent real-time delivery of SCADA messages, which would result in the loss of monitoring information or even of the ability to control entire portions of the SCADA system. Nevertheless, it is foreseen that WSNs will be an integral part of a wide variety of CIs, for a number of reasons [9], which can be grouped in two main categories:

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 225553 (INSPIRE Project).

Luigi Coppolino is with the Dipartimento per le Tecnologie of the University of Naples "Parthenope" (DiT), ITALY (phone:+39 (0)81 5476702; fax: +39 (0)81 5476777; e-mail: luigi.coppolino@uniparthenope.it), and with the Consorzio Interuniversitario Nazionale per l'Informatica (CINI) - ITALY.

Salvatore D'Antonio is with DiT and with CINI (e-mail: salvatore.dantonio@uniparthenope.it).

Luigi Romano is with DiT, CINI, and with the Institute for High Performance Computing and Networking of the Italian National Research Council (CNR -ICAR) (e-mail: luigi.romano@uniparthenope.it).

Gianluigi Spagnuolo is with CINI (e-mail: gianluigi.spagnuolo@gmail.com)

978-1-4244-8081-4/10/$26.00 ©2010 IEEE

Technical:

- WSN technology has the potential of significantly improving the sensing capabilities of SCADA sub-systems, since it provides a wide variety of low-cost sensors, which can be easily and flexibly deployed [10][17].
- The use of WSNs may result in increased resilience of the overall SCADA architecture, due to the capability of sensors to build a mesh-based routing topology [18][23].

Political:

- Governments around the world have recognized the importance of Wireless Sensor Networks as a key technology for the protection of CIs, including Critical Information Infrastructures (CIIs), and have issued formal directives - as well as funded specific programs - for favoring the development of WSN technology in the context of CI protection. The U.S. Department for Homeland Security 2004 National Plan for Research and Development in Support for CIP [11] and the Cooperative Research Center for Security (CRC-SAFE) launched by the Australian government [14], are two remarkable examples in a long list of initiatives aiming at dependable deployment of WSN technology in CIs.

In this context, the availability of efficient Intrusion Detection System (IDS) technology, specifically tailored to protecting the WSN-based areas of a CI, is of utmost importance.

Regrettably, all currently available solutions have serious drawbacks, in terms of performance and/or applicability, as discussed in detail in section 0

The IDS solution that we propose in this paper is the result of a thorough analysis of the specific security issues, as well as of the architectural characteristics and constraints of WSNs, when embedded in a CI. The IDS is organized as a distributed application, consisting of several probes, which are deployed over the wireless clouds (as well as over the wired trunks) of a CI.

We explicitly emphasize some key advantages of our solution over existing ones, and in particular:

- It exploits specific characteristics of the most commonly used protocols for WSNs in SCADA systems (as opposed to many existing solutions, which are a mere porting of traditional IDS solutions for a wired setting).
- It looks at sensors as a main source and/or target of attacks (while in many existing products, this is not the case). We claim that security issues of the sensor level are key, since this level is typically the least protected one (for a number of reasons, and particularly the impossibility of using strong cryptographic techniques, due to the limited computing power available at the nodes, and the prohibitive costs in terms of battery consumption of cryptographic procedures).
- It provides the possibility of correlating information on attacks to the WSN clouds to information on malicious activities in the wired trunks of the SCADA system, which results in deeper understanding of the actual health of the system.

To validate the proposed solution, an experimental campaign has been conducted, on top of a heterogeneous testbed, integrating COTS WSN products [7] and proprietary SCADA devices by a major vendor [12]. The experimental campaign focused on two emerging classes of attacks to WSNs, namely Sinkhole and Sleep Deprivation attacks[3].

The rest of the paper is organized as follows. Section II. provides an overview of the main vulnerabilities of WSNs routing protocols. Section 0 describes the architecture and operation of the proposed IDS solution, and includes a comparison to existing approaches. SectionIV. deals with implementation details, and Section V. presents the testbed setup and discusses experimental results, which confirm the effectiveness of the proposed approach. Finally, Section VI. concludes the paper with some final remarks.

II. Exploitation of WSN routing vulnerabilities

There is a wide variety of routing protocols for WSNs [4]. The most popular ones are Ad-hoc On-demand Distance Vector (AODV) [32] and MultiHop [4].

AODV is a reactive routing algorithm. It is the routing scheme adopted by ZigBee [33][34]. When a WSN node using AODV protocol receives a packet addressed to an unknown destination, it starts to send “route discovery” packets (Route Request message - RREQ). A “route discovery” message is propagated throughout the network until a node finds an entry in its routing table matching the address, and responds to the request with a Route Reply message (RREP). Each message brings the receiving device to set a path in its routing table, if the sender of that message is an in-sight node. When the message reaches its destination, a backward path has been set along the intermediate nodes.

Multihop and its enhancements (e.g., CTP [22], MintRoute [31], and MultiHopLQI [8]), use a shortest path first algorithm, which gives priority to the route to the base station having the lowest cost. The cost function can be based on either the hop count to the base station or on the estimate of the link bandwidth. These values are used to select the parent node, that is the neighbor node with the best path metric. Multihop nodes periodically send route update messages with routing information to their neighbors. These messages contain the measured Expected Transmission cost (ETX) to the base station and a measure of the link quality for every neighbor node. Since this protocol family implements no security features, malicious route updates can be used to perform an attack against a network.

A. Sinkhole attack

A sinkhole attack exploits the vulnerabilities of malicious node which has been previously compromised by an intruder. Starting from the assumption that an attacker gained access to

a sensor node and changed its internal state, the malicious node forges false route packets with a low hop count value to the base station. In this way the malicious node looks very attractive to the surrounding sensors. In other words, the malicious node attracts neighboring nodes with fake routing information, which makes the attacker node a convenient intermediary. As a result, many sensor nodes will attempt to route their traffic through the compromised node. The attacker will thus be able to modify the content of the data packet, drop it, or launch additional attacks (e.g. selective forwarding attack, blackhole attack, and more).

In case the routing protocol being used by the WSN nodes is AODV, then the attacker, which has successfully compromised a node, broadcasts a RREQ for a route to the base station. Afterwards it sends a RREP, which contains the maximum Destination Sequence Number and the minimum hop count. Every node which receives the initial RREQ will either reply to the malicious node (if a route to the base station exists in its routing table), or broadcast the message again and, eventually, will receive the RREP message from the malicious node. This will cause it to assume that the compromised node is along the best path to the base station and to update its routing table accordingly.

Multihop routing protocol (MintRoute, MultiHopLQI, CTP, Xmesh) uses link quality estimates to build the routing table. The attacker sends false route messages, which notify neighbors about his attractive characteristics (low path cost, high link quality, and the like). Then, it intercepts and changes the content of the routing packets so that the other nodes seem to have worse characteristics.

B. Sleep Deprivation

The precondition for a sleep deprivation attack is that a malicious user has gained control of a WSN node (the red node in Fig. 1). Assuming that the precondition is met, two alternative attack techniques can be used:

1. the compromised node forwards the same packets multiple times,

2. the compromised node generates fake packets with a high frequency.

Such techniques have two negative effects on the WSN: i) discharge of batteries of all the nodes along the route (the orange nodes in Fig. 1) from the malicious node to the base station; and, ii) a Denial of Service for those nodes (the yellow nodes in Fig. 1) whose path towards the base station (dashed lines in Fig. 1) cross the attacked, overloaded, path between the attacker and the base station (the red one).

In MultiHop a sleep deprivation attack can be conducted by sending routing packets in broadcast. The attack is amplified if the fake routing packets force some nodes to change their parents, as in this case each fooled node will notify, generating more traffic, the change to all its neighbors.

In AODV the attack may be conducted by sending unnecessary routing requests (RREQ), or by sending forged RREP packets that force the creation of loops in the WSN. In this case, due to the loops, packets are forwarded and stay alive for more time, hence resulting in lots of retransmissions and additional route messages.

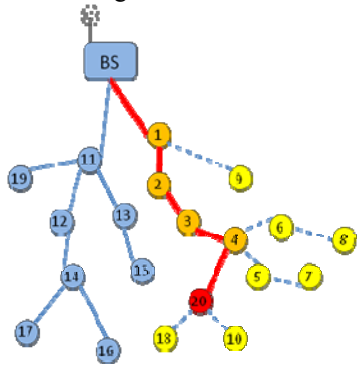


Fig. 1: Sleep deprivation attack effect

III. Architecture and Operation of the Proposed IDS

In this section, we i) analyze the requirements of an IDS for a CI which includes WSN zones, ii) present the main design principles of the solution that we propose, and iii) discuss the main advantages of our solution (as compared to existing approaches). Focus is on security issues of the WSN trunks. A thorough treatment of issues which are more related to the wired part can be found in [20].

IDS solutions proposed for WSNs can be categorized in two main classes: centralized [19][29], and distributed [25][21] solutions.

In centralized solutions, sensor nodes feed an IDS agent running on a host which is connected to the WSN with control data. The IDS agent analyzes the data and possibly detects ongoing attacks. Since routing attacks can prevent control packets from reaching the IDS agent, in centralized approaches the IDS agent may get to an erroneous view of the network, and ultimately fail to detect the attack. This is one of the major drawbacks of centralized approaches.

In decentralized solutions, it is the sensor nodes which run the logic for detecting the attacks. As such, distributed solutions are potentially more resilient to network level attacks, since it is still possible that attacks be detected locally, even in the event of severe damage to the network infrastructure (although in this case there is a risk of the IDS system getting to a wrong decision, due to a non consistent view of the global status). Additionally, distributed solutions need the execution of agreement protocols to allow each node to share its local view of the network with a set of neighbors. This results in consumption of additional resources, mainly due to an increased number of transmissions.

Another typical classification of IDS solutions is based on the kind of analysis which is used to spot an attack. Two classes of IDSs are available: misuse based, and anomaly based. In misuse based IDSs, the detection agent has a knowledge base, in the form of attack signatures, which characterize the kinds of (known) malicious activities that the attacker may conduct. At runtime, the detection agent compares the profile of the current traffic to the signatures of the known attacks. If a match is found, an alarm is raised. In anomaly based solutions, the detection agent is trained with what is assumed to be the “normal” behavior of the system,

(typically represented through a mathematical model), and compares the actual behavior of the system against the normal profile: any deviation from the normal behavior is considered as a potential attack. Misuse based solutions are typically more accurate (particularly because they typically have much lower false positive rates), but they are unable to detect the so called zero day attacks (and more in general any previously unknown attack). On the other hand, anomaly based solutions can potentially achieve present a higher detection rate (since they can spot virtually any kind of attack), but they typically exhibit a higher number of "false positive" (i.e. normal traffic recognized as an attack).

We propose a hybrid detection solution where any node runs a detection agent which is in charge of identifying suspicious nodes. Suspicious nodes are inserted temporarily in a blacklist and an alarm is sent to the central agent. The final decision is demanded to the central agent.

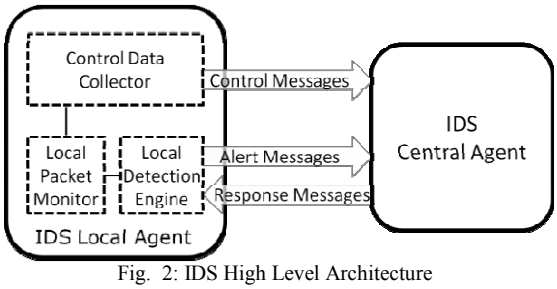


Fig. 2: IDS High Level Architecture

As for the IDS agents running on the wireless sensors, the following design constraints apply, which reflect specific characteristics of WSNs [28]:

- Low memory footprint: agents must use the least possible amount of memory;
- Low CPU usage: agents must use a simple algorithm, to minimize CPU load and power drain;
- Low network footprint: since communication is power consuming, agents must minimize the number of messages which are exchanged.

Fig. 2 shows the architecture of our hybrid IDS in detail. The solution we propose combines misuse and anomaly based techniques in a two-level distributed hierarchy for improved resilience and performance. The **IDS Local Agent** (LA), running on a sensor node, is made of: i) the **Local Packet Monitor** that is in charge of analyzing the traffic flowing through the node; ii) the **Control Data Collector** that gathers measures to be sent to the **IDS Central Agent** (CA); iii) the **Local Detection Engine** that is in charge of: a) detecting suspicious activities and rising alerts; b) receiving responses from the CA; and c) performing possible recovery actions. The Local Detection Engine works by identifying anomalous events with respect to measures taken by the Control Data Collector. The occurrence of specific combinations of such events is taken as a "weak" detection of an ongoing attack and triggers temporary reaction actions. The usage of an anomaly based approach for local detection might results in a high number of false positives. The usage of a temporary local decision allows for mitigating such side effect and especially avoiding the triggering of reactions for intermittent anomalies. The temporary decisions taken by the LA can be made persistent by the CA which is in charge of recognizing attacks by exploiting control data and alarms sent by LAs. In particular, it detects attacks based on patterns of attack features (misuse based detection). When the CA makes its final decision, the base station propagates the decision to the LAs for its enforcement.

Specifically, when an alert is raised by the Local Detection Engine, the LA performs the following actions:

1. adds the suspicious node to a *blacklist*;
2. changes the parent by choosing one of the nodes not in the blacklist as new parent;
3. sends an alert message to the CA containing the suspicious events;
4. waits for a response from the CA; if such a response is missing the LA assumes that the attacker is preventing the alert from reaching the CA and forces another run starting from point 1 until either a notification of the decision made by the CA is received or all the neighbours have been assessed as parent. In the second case the blacklist is empted and the process is restarted since, due to local reactions, a new path toward the BS should be established.
5. If the CA does not confirm the attack, the LA empties the black list and returns to regular mode. Else if the CA decision confirms an ongoing attack, the LA freezes the blacklist waiting for a reaction activity suggested by the CA.

IV. Implementation Details

A. Implementation of the Attacker

The malicious node runs an attack injection tool, which implements a modified version of CTP to perform either a sinkhole attack or a sleep deprivation attack.

While conducting a sinkhole attack the malicious node pretends to have a very low Expected Transmission cost, and modifies the value of its LEEP (Link Estimation Exchange Protocol) packets, both values are used by neighbor nodes to estimate the bidirectional link quality. Since the CTP is a conservative protocol in the sense that it tries to preserve already established paths, the attacker has to transmit fake values with a very high frequency in order to be able to force changes in the routing tables of the victim nodes.

As for the sleep deprivation, we implemented the attack by forwarding the same packets multiple times. In CTP-based implementation of the sleep deprivation attack, nodes store received packets in an internal queue (*last received packets queue*) which is scanned in order to detect and discard duplicate packets. To avoid its fake packets being discarded the attacker retransmits them with a certain latency. In fact, as the *last received packets queue* of the victim nodes have a fixed length (say N), after the victim node has received N non-duplicated packets all the previous fake-packets are shifted-out of the *last received packets queue* and the node is no more able to recognize an older packet as a duplicate.

B. Implementation of the IDS Local Agent

The IDS Local Agent (LA) collects control data by using either specific CTP utility interfaces (CtpInfo, CtpPacket) or ad-hoc counters added in the send and forward event handlers and in the link estimator sub-system.

Nodes can automatically transmit status information to the base station. Examples of such information include the radio traffic, parent node and neighbors, number of generated data messages, number of generated routing messages, number of forwarded messages, number of dropped messages, number of retransmitted messages, etc. Instead of measuring the exact values of such parameters, the LA estimates their average values. Average values are computed in the form of Exponential Moving Averages (EMAs) with the following formula:

$$EMA_{t(i)} = EMA_{t(i-1)} + \alpha\,(M_{t(i)} - EMA_{t(i-1)})$$

$$\alpha = 2/(N+1)$$

where $EMA_{t(i)}$ is the value of the average at time i, $M_{t(i)}$ is the new value of the considered parameter, and N is the length of the sliding window considered for the evaluation of the average.

The motivation for using EMAs instead of moving averages is twofold: i) in order to evaluate the EMA at time i the node must only store the value of the EMA at time i-1 (this allows for reducing the memory space used in the node); ii) EMA is more reactive to changes of the measured parameter than the moving averages, thus resulting in a lower detection latency.

A node generates an alert for one of the monitored parameters, if its EMA value falls outside a reference interval.

Depending on the anomalous parameter the IDS LA considers some of the neighbors as suspicious. After a node is recognized as suspicious, it is inserted into a blacklist. The *blacklist* is implemented by adding a flag column to the existing CTP neighbour table stored in each node.

C. Implementation of the IDS Central Agent

The CA combines the alerts obtained from LAs to decide if they must be confirmed or discarded. The criteria behind alert correlation are attack specific. In the implementation of the IDS Central Agent we focused on a sinkhole attack. In this case we assume that there is an ongoing attack if the majority of the neighbors of a node have raised alerts for anomalous routing parameters, like BRVC (number of beacon packets received) and SND (number of beacon packets sent).

Fig. 3 shows the dynamic of the EMA values for the following metrics: number of beacon packets sent (bsnd), number of beacon packets received (brcv), number of data packets sent (snd), number of forwarded packets (fwd), number of dropped packets (drp), number of retransmitted packets (rtr), number of not expected packets (alien).

The EMAs values are monitored for an attacker (the first graph) and three neighbor victim nodes (graph from the second to the fourth). At time 25 the steady state is reached. At time 32 the second node shows an anomaly (it got a new child) and its LA raises an alert. The CA will discard such an

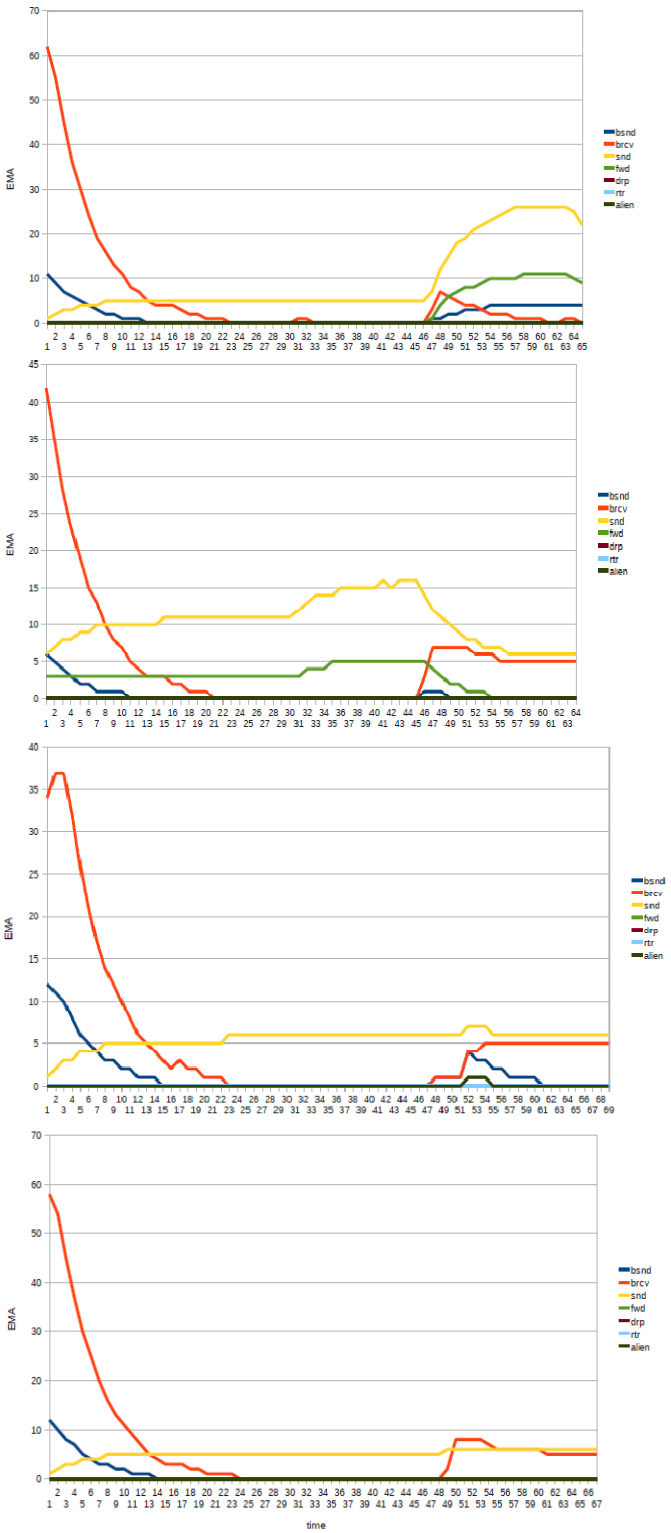


Fig. 3: Key performance indicators

alert since just one out of three nodes is misbehaving. At time 45 the malicious node launches a sinkhole attack. All the neighbor nodes show multiple anomalies, hence resulting in alerts being thrown. In this case the CA confirms the alert and gives evidence of an ongoing attack.

V. EXPERIMENTAL RESULTS

Our experimental campaign was conducted using the INSPIRE project testbed shown in Fig. 4. The testbed consists of wired sensors, which are connected to a supervisory station through Remote Terminal Units (RTUs), an ad-hoc RTU, which has been developed to connect wireless sensors to the rest of the SCADA system via a programming board, i.e. a base station, a legacy RTU manufactured by Elsag Datamat [12], which receives data from wired field sensors, a legacy GLAW SCADA server developed by Elsag Datamat, an iFIX RTU [13], and a supervisory station running an iFIX SCADA server.

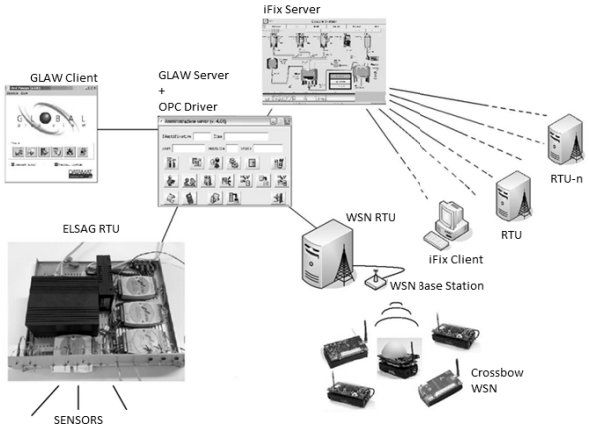


Fig. 4: INSPIRE testbed setup

The wireless sensor network is composed of several CrossBow [7] IRIS motes and a CrossBow MIB520 USB programming board. Each IRIS mote is equipped with an Atmega128 processor, an Atmel RF230 (IEEE 802.15.4 compliant) radio transceiver, and a CrossBow MDA100 light and temperature sensor board. Sensors run TinyOS 2.x and use CTP.

We conducted our experiments on a simulated WSN. The simulation was performed by using TOSSIM [27]. Each simulation run was conducted with a number of node ranging between 100 and 250 and the following settings: only one attacker, nodes are stationary, and a number of neighbors for each node between 5 and 10. The results of the simulation tests were confirmed by executing an experimental campaign on a real network consisting of a number of nodes ranging from eight to fifteen. At each run of the test on the real network we physically deployed the nodes following widely accepted best practices for WSN deployment [1][2], in order to avoid common pitfalls (e.g., configurations with single points of failure - i.e. nodes whose failure results in the partitioning of the network - were avoided).

The campaign was conducted following the workflow shown in Fig. 5.

The workflow includes four phases. Sensor nodes periodically generate control messages whose content is processed by the IDS CA module (Phase 1) to detect possible ongoing attacks and identify the attacker.

During Phase 2 the malicious behavior of the attacker node is triggered.

During Phase 3, the nodes under attack: i) recognize an anomalous behavior, ii) put the suspicious nodes in a black list, and iii) generate alert messages. In Phase 4, alerts are sent to the IDS CA, which in turn - based on the control messages which have been gathered and the received alerts - makes its decision. It is worth noting that the IDS CA continuously checks control packets also in absence of alerts. Hence it is able to detect possible attacks evading detection performed by the Local Agents.

Experiments were conducted focusing on the sinkhole attack. To compare our hybrid solution to a purely centralized solution, we deployed two different scenarios.

In scenario A we assume that the malicious node modifies sensor data, but it does not modify the control messages which are used for detection purposes. In scenario B, we assume that the attacker is smarter, and it is able to modify all received messages (data and control packets).

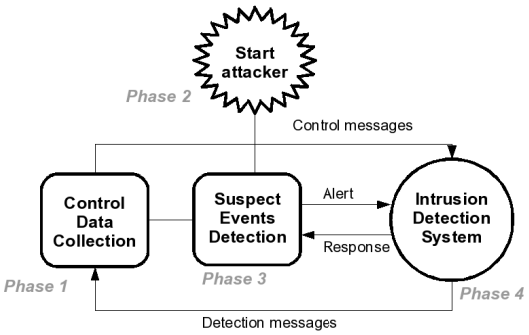


Fig. 5: Experimental campaign workflow

A first test session was executed with a simulated network made of a random number of sensors. In this session we tested 100 different topologies each in 10 different simulations. In each run a random node behaves as the attacker. The results of the simulation tests have been validated by comparing them to those attained in the real network environment. Table 1 shows the average values of detection rate. It is quite evident that in a simplified condition (Scenario A) both approaches provide optimal results. Conversely, when the attacker is smarter, the centralized IDS is ineffective, whereas our hybrid approach still performs well.

TABLE 1
DETECTION RATE RESULTS (AVERAGE NEIGHBORHOOD SIZE = 8)

	Scenario A	Scenario B
Centralized	94%	< 20%
Hybrid	95-97%	93-96%

The detection rate of our hybrid approach is also comparable to the performance of a distributed solution in the configuration environment of scenario B. As a reference study, the detection rate for routing attacks is, between 90% and 95% depending on the consumed memory [20].

Given a certain detection rate, it is worth comparing the resulting number of false positives. In [21], when the detection rate of the distributed solution is comparable to the detection rate of our hybrid solution, the false positives ratio ranges

from 6 to 10 percent. In our solution we had a percentage of false positive of about 3%.

In terms of memory usage, the size of the IDS agent of the hybrid system is comparable with the fingerprint of the agents of a fully distributed systems running TinyOS [25][26], as shown in Table 2.

TABLE 2
FINGERPRINT OF DISTRIBUTED AND HYBRID IDS AGENTS

	Krontiris 2009	Krontiris/LIDeA	Hybrid
RAM	583 bytes	808 bytes	734 bytes
ROM	9427 bytes	10046 bytes	3208 bytes

VI. CONCLUSIONS

In this paper we presented an Intrusion Detection System for protecting Critical Information Infrastructures using Wireless Sensor Network technology.

The proposed system relies on a hybrid detection approach in the sense that any node runs a detection agent which is in charge of identifying suspicious nodes. In order to validate the system an experimental campaign has been conducted, which demonstrated the effectiveness of the proposed approach against some emerging attacks to WSNs, namely sinkhole attacks and sleep deprivation attacks. Also importantly, results demonstrated that our solution satisfies the stringent requirements (in terms of limited availability of resources) which are typical of Wireless Sensor Networks.

VII. REFERENCES

Periodicals:

[1] Akyildiz, I.F., Weilian Su, Sankarasubramaniam, Y., Cayirci, E., A survey on sensor networks, Communications Magazine, IEEE , vol.40, no.8, pp. 102-114, Aug 2002

[2] Stajano, F., Hoult, N., Wassell, I., Bennett, P., Middleton, C., and Soga, K. (2010) : Smart Bridges, Smart Tunnels: Transforming Wireless Sensor Networks from Research Prototypes into Robust Engineering Infrastructure, Ad Hoc Networks, In Press, Corrected Proof, Available online 16 April 2010, ISSN 1570-8705, DOI: 10.1016/j.adhoc.2010.04.002

Books:

[3] Cayirci E., Rong C.: "Security Attacks in Ad Hoc, Sensor And Mesh Networks" in Security in Wireless Ad Hoc, Sensor and Mesh Networking, John Wiley & Sons, 2008, ISBN 978-0-470-02748-6., p107-120

[4] Sajal K. Das and Habib M. Ammari, "Routing and data dissemination in wireless sensor networks", Invited Book Chapter, Wireless Sensor Networks: A Networking Perspective (J. Zheng and A. Jamalipour, Editors), Wiley-IEEE Press, July 2009

Technical Reports:

[5] Beech, E.: Cyberspies penetrate electrical grid: report. Reuters top ten news stories, Wed Apr 8, 2009. [Online] Available: http://www.reuters.com:80/article/topNews/idUSTRE53729120090408 (last accessed 01/06/2010)

[6] Critical Foundations: Protecting America's Infrastructures. The Report of the President's Commission on Critical Infrastructure Protection. [Online] Available at http://www.fas.org/sgp/library/pccip.pdf (last accessed 31/05/2010)

[7] CrossBow. http://www.xbow.com (last accessed 01/06/2010)

[8] Fonseca R., Gnawali O., Jamieson K., and Levis P.: TinyOS Enhancement Proposals 119: Collection. 09-Feb-2006. [Online] Available: http://www.tinyos.net/tinyos-2.x/doc/html/tep119.html (last accessed 01/06/2010)

[9] Roman R., Alcarza C., and Lopez J.: The role of wireless sensor networks in the area of critical information infrastructure protection, Information Security Technical Report Vol. 12 , Issue 1, pages 1363- 4127, 2007.

[10] Ye, W., Heidemann, J.: Enabling Interoperability and Extensibility of Future SCADA Systems. Technical Report ISI-TR-625, USC/Information Sciences Institute, November, 2006

[11] U.S. Government, 2004. US National Plan for Research and Development in Support for CIP. April 8, 2005. Online: http://www.dhs.gov/xlibrary/assets/ST_2004_NCIP_RD_PlanFINALApr05.pdf (last accessed 02/06/2010).

[12] http://www.elsagdatamat.com/EN/Automation.htm (last accessed 02/06/2010)

[13] http://www.ge-ip.com/products/3311 (last accessed 12/06/2010)

Papers Presented at Conferences (Unpublished):

[14] Bopping D. CIIP in Australia. In: First CI2RCO critical informationinfrastructure protection conference, Rome; March 2006.

[15] Byres, E., Eng, P.: Who Turned Out The Lights? Security Testing for SCADA and Control Systems. CanSecWest Conference 2006.

[16] Peterson, D.: Intrusion Detection and Cyber Security Monitoring of SCADA and DCS Networks. Presented at ISA Automation West. 2004

[17] Wolmarans, V., Hancke, G.: Wireless Sensor Networks in Power Supply Grids. SATNAC 2008, Wild Coast Sun, September 2008

Papers from Conference Proceedings (Published):

[18] Bai, X., Meng, X., Du, Z., Gong, M., Hu, Z.: "Design of Wireless Sensor Network in SCADA System for Wind Power Plant" in *Proceedings of the IEEE International Conference on Automation and Logistics*, Qingdao, China, September 2008

[19] Bo Y.; Bin X.: Detecting selective forwarding attacks in wireless sensor networks. IPDPS 2006. 20th International Parallel and Distributed Processing Symposium, 2006., vol., no., pp. 8 pp.-, 25-29 April 2006

[20] Coppolino, L., D'Antonio, S., Esposito, M., Romano L.: Exploiting diversity and correlation to improve the performance of intrusion detection systems. Proceedings of the First International Conference on Network and Service Security, Paris, France, June 2009.

[21] da Silva, A. P., Martins, M. H., Rocha, B. P., Loureiro, A. A., Ruiz, L. B., and Wong, H. C.: 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 (Montreal, Quebec, Canada, October 13 - 13, 2005). Q2SWinet '05

[22] Gnawali O., Fonseca R., Jamieson K., Moss D., and Levis P.. Collection Tree Protocol. In Proc. of the 7th ACM Conf. on Embedded Networked Sensor Systems (SenSys), 2009.

[23] He, Z.Y., Zhang, J., Li, H.W., Bo, Z.Q., Zhang, H.P., Nie, Q.W.: An Advanced Study on Fault Location System for China Railway Automatic Blocking and Continuous Transmission Line. IET 9th International Conference on Developments in Power Systems Protection (DPSP 2008)

[24] Hwang S., "Frequency domain system identification of helicopter rotor dynamics incorporating models with time periodic coefficients," Ph.D. dissertation, Dept. Aerosp. Eng., Univ. Maryland, College Park, 1997.

[25] Krontiris, I., Giannetsos, T., and Dimitriou, T.: LIDeA: a distributed lightweight intrusion detection architecture for sensor networks, in Proceedings of the 4th international Conference on Security and Privacy in Communication Networks (Istanbul, Turkey, September 22 - 25, 2008). SecureComm '08

[26] Krontiris, I., Benenson, Z., Giannetsos, T., Freiling, F. C., and Dimitriou, T.: Cooperative Intrusion Detection in Wireless Sensor Networks. In Proceedings of the 6th European Conference on Wireless Sensor Networks (Cork, Ireland, February 11 - 13, 2009). U. Roedig and C. J. Sreenan, Eds. Lecture Notes In Computer Science, vol. 5432. Springer-Verlag, Berlin, Heidelberg, 263-278.

[27] Levis, P., Lee, N., Welsh, M., and Culler, D. 2003. TOSSIM: accurate and scalable simulation of entire TinyOS applications. In *Proceedings of the 1st international Conference on Embedded Networked Sensor Systems* (Los Angeles, California, USA, November 05 - 07, 2003). SenSys '03. ACM, New York, NY, 126-137.

[28] Martynov, D.; Roman, J.; Vaidya, S.; Huirong Fu, Design and implementation of an intrusion detection system for wireless sensor networks. Electro/Information Technology, 2007 IEEE International Conference on , vol., no., pp.507-512, 17-20 May 2007

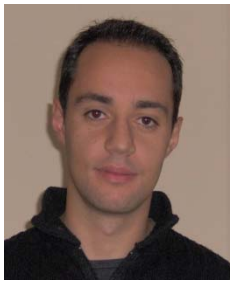
[29] Ngai, E. C., Liu, J., and Lyu, M. R., Comput: An efficient intruder detection algorithm against sinkhole attacks in wireless sensor networks. Commun. 30, 11-12 (Sep. 2007), 2353-2364
[30] Verba, J., Milvich, M.: Idaho National Laboratory Supervisory Control and Data Acquisition Intrusion Detection System (SCADA IDS). Technologies for Homeland Security, 2008 IEEE Conference on , vol., no., pp.469-473, 12-13 May 2008
[31] A. Woo, T. Tong, and D. Culler. Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys'03)*, Los Angeles, CA, USA, November 2003.

Standards:

[32] C. Perkins , E. Belding-Royer , S. Das, Ad hoc On-Demand Distance Vector (AODV) Routing, RFC Editor, 2003
[33] IEEE 802.15 WPAN Task Group 4, http://www.ieee802.org/15/pub/TG4.html (last accessed 01/06/2010)
[34] ZigBee Alliance, http://www.zigbee.org (last accessed 01/06/2010)

VIII. BIOGRAPHIES

Luigi Romano is currently an Associate Professor at the University of Naples "Parthenope". His research interests include some of the fundamental aspects of computer system dependability, namely availability, reliability, performability, and security. He has been at the Center for Reliable and High-Performance Computing of the University of Illinois at Urbana Champaign for about two years, doing research with Prof. R.K. Iyer. He has also worked as a consultant for Ansaldo Trasporti and Ansaldo Segnalamento Ferroviario in the field of safety critical computer systems design and evaluation. He received his MS degree in Electronic Engineering and Ph.D. degree in Computer Science from the University of Naples Federico II.



Luigi Coppolino is an assistant professor at the University of Naples "Pathenope". He got a Ph.D. degree in Computer Engineering at the University of Naples Federico II. His research activity mainly focuses on critical networked systems. He is/was one of the principal investigators of various European Research projects including the INcreasing Security and Protection through Infrastructure REsilience (INSPIRE) and the INSPIRE International Cooperation projects. In 2006 the TrustedSIM, javacard based project, developed by Mr. Coppolino, was one of the winners of the 7th SIMAGINE contest, sponsored by Axalto, SUN Microsystems and Samsung.

Salvatore D'Antonio is currently an Assistant Professor at the Department of Technology of the University of Naples "Parthenope", in Italy. His current research interests include network monitoring and management, intrusion detection systems and information system security. He is currently the Project Coordinator of the EU-funded FP7 INSPIRE and INSPIRE-International projects.

Gianluigi Spagnuolo is currently completing a Master degree program in Critical and Networked Systems at the University of Naples Parthenope. He got a Bachelor degree in Computer Engineering from the University of Naples Federico II. Mr. Spagnuolo has published many articles in national magazines on security. He had two Summer of Code Google experiences. Today Mr. Spagnuolo has an intense collaboration with the FITNESS research group of the University of Naples Parthenope in the framework of the INSPIRE project.