An Intrusion Detection System for Wireless Process Control Systems*

Tanya Roosta1, Dennis K. Nilsson2, Ulf Lindqvist3, Alfonso Valdes3
Dep. of Electrical Engineering and Computer Science, UC Berkeley 1
Department of Computer Science and Engineering, Chalmers University of Technology2
Computer Science Laboratory, SRI International3
roosta@eecs.berkeley.edu1, dennis.nilsson@ce.chalmers.se2, {ulf.lindqvist,alfonso.valdes}@sri.com3

Abstract

A recent trend in the process control system (PCS) is to deploy sensor networks in hard-to-reach areas. Using wireless sensors greatly decreases the wiring costs and increases the volume of data gathered for plant monitoring. However, ensuring the security of the deployed sensor network, which is part of the overall security of PCS, is of crucial importance. In this paper, we design a model-based intrusion detection system (IDS) for sensor networks used for PCS. Given that PCS tends to have regular traffic patterns and a well-defined request-response communication, we can design an IDS that defines the model of normal behavior of the entities and detects attacks when there is a deviation from this model. Model-based IDS can prove useful in detecting unknown attacks.

1. Introduction

We present a design of a distributed, multi layer, model-based intrusion detection system for process control systems combining legacy control components with emerging sensor network technology. Modern industrial processes in energy and manufacturing increasingly depend on digital monitoring and control, such as Supervisory Control and Data Acquisition (SCADA) and distributed control system (DCS) in the electric power and oil and gas sectors. An important trend in such systems is the use of wireless sensor networks, which allows establishment of a cost-effective and highly reliable network and communication infrastructure. Such systems have been fielded in large refineries, where the geographical extent is on the order of kilometers, as well as pipelines, where the extent can be hundreds of kilometers. Integrating sensor nets into SCADA and DCS enables much finer grained process monitoring and permits access to locations where wired connections are difficult or expensive. However, this use of sensor networks may introduce vulnerabilities beyond the vulnerabilities present in SCADA or DCS. Monitoring the network itself (to include both the wireless sensor network portion and the conventional SCADA portion) is essential to detect and mitigate attacks. To be very careful about the terminology, SCADA is commonly used as an umbrella term for process control systems. A SCADA system monitors and controls a geographically dispersed process like an oil pipeline, while DCS monitors and controls a local process such as an oil refinery [8]. Because many components and applications of SCADA and DCS are similar or identical, we use the term SCADA throughout this paper.

A major drawback of typical SCADA systems is their inflexible, static, and often centralized architecture, accompanied by the high cost of wiring devices together. Wireless sensor networking is a promising technology that can significantly improve the sensing capability of the SCADA system and significantly reduce the wiring cost. SCADA systems are now leading the way for better control and monitoring of remote processes and plants through the use of new cost-effective hardware and software. Recently, a number of companies, such as Dust Networks and Emerson Process Management, have teamed to bring sensor networks to the field of process control systems. Their solutions offer a reliable, robust, and secure sensor network that could interoperate with the existing wired and wireless devices of SCADA systems. In this paper, we focus on SCADA systems in the oil and gas industry, in particular pipelines. Better control and monitoring is a real asset for oil and gas pipelines due to facilitating the distribution. Therefore, the oil and gas industry is moving in the direction of using wireless sensor networks in its deployment to reduce costs and increase data gathering reliability.

The rest of the paper is organized as follows: in Sec-

*This material is based upon work supported by the US Department of Homeland Security under Contract Number HSHQDC-07-C-00006. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the US Department of Homeland Security.
tion 2 we discuss why the security aspect of process control systems is crucial. Section 3 describes the components of a typical SCADA system. Section 4 discusses the WirelessHART specification. Section 5 describes the threat model for sensor networks used in SCADA systems and discusses the assumed capabilities of the attacker. Section 6 gives a detailed description of our proposed model-based IDS, followed by Section 7 that describes a set of implementable policy rules for our IDS.

2. Motivation

The security of SCADA systems is crucial given the criticality of their application. A successful cyber attack on SCADA and other control systems could result in massive cost and potential loss of life. Security studies from the U.S. Department of Energy (DOE) and commercial security consultants have demonstrated the cyber vulnerabilities of control systems. More than 60 identified (though not publicly documented) real-world cases have occurred where electronic means have impacted the control systems’ reliable operations. Control systems have been designed with specific reliability and availability requirements but not specific cyber security requirements. Adapting cyber security to these requirements is another challenge, and one that must be met by the system designers. If malicious actors could access these systems, they would have access to operational data critical to the operation of the system. Also, a knowledgeable attacker could modify the data used for operational decisions, the programs that control critical industry equipment, or the data reported to control centers. The impact could be destructive. Such attacks could exceed equipment design and safety limits and potentially result in damage, premature system shutdown, and interference with safety system operations. Or they could immobilize control equipment. Consequences could include endangerment of public health and safety, environmental damage, or significant financial impacts due to loss of power production, transmission, or distribution. Some of the possible attacks on these systems are [4, 2, 1]: Gain access to the SCADA system, Identify the devices on the network, Disrupt the master-slave communication, Disable the slave, Read and write data to the slave, Program the slave, Compromise the slave, Disable the master, Write data to the master, and Compromise the master.

Control systems are susceptible to attack because they were not designed to meet cyber threats. As control systems move from traditional closed networks into highly interconnected heterogeneous networks, containing both standardized and legacy technologies, the threat environment changes. In light of these heterogeneous networks, we cannot directly apply security technologies designed for common business IT systems to control systems and still provide adequate protection. Very little work has been done in the area of intrusion detection for SCADA systems, and to the best of our knowledge. No work has been done in the area of intrusion detection for sensor networks used in SCADA systems. In this paper we propose a design for a model-based intrusion detection system for SCADA systems. However, at the current time, we have not implemented our proposed solution. We use the WirelessHART [3] as the basis for our IDS. The details of this standard are given in later sections.

3. SCADA Architecture

The general composition of the SCADA system includes input and output signal hardware, controllers, human machine interface (HMI), networks, communication, databases, and software. The bulk of the data acquisition comes from the remote terminal units (RTUs). An RTU consists of intelligent field devices, for example, controller devices or wireless devices. In the sensor network deployment, the wireless devices are sensor motes, as shown in Fig. 1. The IFDs can be queried by handheld devices in the field. The data collected by an IFD is then compiled and formatted in such a way that a control room operator using the HMI can make appropriate supervisory control decisions. The other devices in the field are gateways that connect the RTU with the process automation network. Communication between the control devices and data acquisition devices generally follows a master-slave protocol, meaning that the slave devices respond only when being polled by the master device. However, in an unusual event, notification by exception is also allowed. SCADA systems have a number of characteristics which distinguish them from the typical IT system. For example, availability and real-time response
take precedence over confidentiality of the data, and the system has a long-lifetime requirement. These characteristics result in different prioritization of the security requirements in SCADA networks. Therefore, the main goal of the attacker is then to compromise the availability and integrity of the gathered data.

4. Background

WirelessHART is a trademark of the HART Communication Foundation (HCF). WirelessHART is the first open wireless communication standard designed to address the critical needs of the process industry for reliable, robust, and secure wireless communication in real-world industrial plant applications. WirelessHART is targeting the process automation, process automation systems, and plant asset management industries. WirelessHART uses a wireless mesh network, meaning that all the field devices can perform routing. In addition, the redundant paths make the network reliable since the messages can route around obstacles and hot spots. Other advantages of WirelessHART include [3]: 1) Compatibility with the existing wired devices running the HART protocol, 2) Frequency hopping for added reliability and minimizing interference, 3) Multiple transmit power levels, which can be configured by the network user, 4) Clear channel assessment, 5) Multiple message modes, such as event notification, block data transfer, and acyclic request-response, and 6) Security through using the AES-128 ciphers and keys. The network manager resides on the gateway(s) and coordinates communication schedules, key management, and monitoring of physical health reports. Inter-node communication, which is used solely for routing, is performed at the NETWORK layer.

5. Threat Model

We outline the threat model used in our analysis. We do not try to model the threats related to the existing legacy SCADA systems, but specifically focus on the threat model in sensor networks. The nature of wireless networks makes them very vulnerable to malicious attacks. First of all, the fact that these networks use wireless links makes them susceptible to various attacks ranging from passive eavesdropping to active interfering. In addition, unlike wired networks, gaining physical access to wireless networks is comparatively easy, and the attacks can target any node in the network. The potential damages include information leakage from the network communication, message contamination, and node impersonation. Attacks on sensor networks can be put into different general categories [7]: 1) A node-class attacker vs. a laptop-class attacker, 2) An insider attacker vs. an outsider attacker, 3) Passive vs. active attacker.

This threat model makes sense in the context of SCADA systems. A powerful attacker can potentially compromise a set of nodes thus becoming an insider attacker (access to the cryptographic keys). Physical security in SCADA systems is lacking. As a result, it is relatively easy to extract cryptographic keys by either physically using a probe on the sensor hardware or by performing power differential analysis. An attacker who modifies the software running on the sensors to send bad observations can be an active attacker. An attacker who does not have access to the cryptographic keys is an outsider attacker, but could be an active attacker by jamming the network. In this paper, we assume that the attacker has the capability to jam the signal as well as physically compromise a subset of the sensor nodes and extract the cryptographic keys. This assumption leads to the following possible threats, which are used to develop the rules/policies of our proposed IDS in Section 7: Threat 1: signal jamming, Threat 2: physical node compromise, Threat 3: denial of service by flooding the network or dropping packets, Threat 4: injection of malicious packets, Threat 5: modification of the en route packets, and Threat 6: replay of transmitted packets.

6. Multi Layer Model-Based IDS

We propose using a model-based intrusion detection system as opposed to a signature-based intrusion detection system. The most important task in building this type of an intrusion detection system is defining a feature space that can accurately represent normal and anomalous behaviors. Our IDS defines normal behavior at different layers of the network stack. The argument in favor of the model-based approach is that there is currently very little information on real attacks in sensor networks. A number of attacks have been proposed with corresponding solutions\(^1\). However, there is no hard information on whether these attacks can be carried out. Especially which ones of those attacks can be used in the SCADA application to disrupt its functionality. As a result, it is very difficult to design an IDS that would require a signature of the exploit or attack. Multi layer in the context of our solution refers to an IDS that defines normal behavior at different layers of the networking stack, such as the physical layer, link layer, and network layer.

6.1. IDS Architecture

Our IDS consists of two components: a central IDS, and multiple IDSs distributed in the field among the sensor nodes, which we call field IDS in the rest of the paper. The main IDS resides on the network manager and is responsible for monitoring all the packets that arrive from the

\(^1\) A comprehensive list can be found in [9].
sensors and are destined for the masters. In addition, this central IDS monitors the traffic from other connected networks, such as the Internet; however, the focus of this paper is on the sensor network portion of the system. We emphasize that our work does not focus on designing rules for intrusions from the Internet since the traffic from the Internet requires a separate set of rules for intrusion detection. The rules we develop in this paper are specifically for sensor network traffic and are not suitable for Internet traffic. The field IDSs are deployed using ‘supernodes’, which are sensor nodes with higher communication and computation power and have tamper-resistant hardware. The reason for choosing supernodes instead of regular sensor nodes for a field IDS is that supernodes are harder to attack and to extract the cryptographic keys from. The field IDSs are responsible for passively monitoring the communication of the sensor nodes in their neighborhoods to collect trace data. They periodically send monitoring messages to the central IDS, where the messages contain information on the traffic patterns of sensor nodes in their vicinity.

We chose this hybrid IDS architecture mainly because of the characteristics of the SCADA system. Fully-distributed IDSs have been proposed in wireless sensor networks, such as [5]. However, if each node were to run an IDS, it would have had to spend more energy monitoring its neighbors and drawing inferences based on its observation. It is also much more complicated trying to arrive at a global decision when using a fully distributed IDS system. The nodes have to pass information among themselves to arrive at a consensus. The fully distributed scheme is also more vulnerable to attacks since the software is running on the nodes that are not tamper resistant. As mentioned in Section 5, it is fairly easy to compromise a subset of the nodes physically and subvert their operation. This will nullify the whole point of using an IDS to monitor the behavior of the sensor nodes. Operation of the central and field IDS consists of three phases:

Data gathering: The field IDS listens in the promiscuous mode to the sensor traffic in its neighborhood and gathers information. The sensors send physical health monitoring reports periodically, for example every 15 minutes. Then, the field IDS extracts relevant information, such as the packet fields from these reports. In addition to the information gathered by the field IDS, the central IDS gathers trace data from the incoming packets (from the gateways) and the field IDSs. This provides the overall IDS with more information given that the central IDS has a global view of the network. This information is used in the next phase. For analysis purposes we have several fields in the physical health reports. The reports contain information about at most N neighbors that have been heard in the last 15 minutes. This includes neighbors that are part of normal scheduled communication, and neighbors that are overheard during the infrequent ‘hello world’ broadcasts that every node makes. The information about each neighbor includes the number of TX attempts, number of ACKs received, number of RXed packets, and number of bad CRC and bad MIC packets received.

Conformity check: Both IDS modules use the extracted data from phase 1 to perform analysis and check for conformity of the node behavior to the normal behavior.

Inference: After the analysis is done, the result is sent to the inference unit for deciding whether a detected anomaly is a transient network failure or a malicious attack. For this phase to make accurate inferences, the IDS must keep a ‘state’ or history for each node it is monitoring to ensure that the system is capable of distinguishing occasional network failures from real attacks, at some confidence level.

The IDS also needs to monitor for lying nodes. Given that the sensor nodes are unattended in the field, it is conceivable that some of the nodes could be physically compromised. An attacker who has access to a node can send wrong/bad observations to the data-gathering station. Therefore, it is important that the IDS checks for invalid or anomalous observations. This can be achieved by comparing the data gathered from clusters of sensor nodes in the same physical neighborhood. The network manager has a global view of the network topology, and therefore is capable of partitioning the sensor observations into neighborhood clusters. Our IDS monitors the network at various levels: the physical layer, link layer, traffic pattern, and communication datagram. In the following subsections, we will explain each of these in more detail, but first we state our assumptions regarding the communication among different IDS components.

We assume that the communication between the central IDS and the field IDS is encrypted using an end-to-end network key to ensure confidentiality and integrity of the messages. In addition, the packets transmitted among sensor nodes have their payload encrypted and have a message integrity code (MIC) on their headers. However, the packet header is not encrypted so that the field IDS can listen to the sensor node communication in promiscuous mode. Moreover, we assume that there is a mechanism in place for authenticating the field IDSs to the network administrator if there is the need to do so, such as using a digital signature.

6.2. Physical Layer

Jamming is the interference with the radio frequency (RF) used by the nodes in a network. It makes use of the broadcast nature of the communication medium. The sensor nodes generally transmit at a pre configured power level. In some scenarios, the nodes can adjust their transmission power to different prescribed levels, but the levels are still known to every node and the administrator since it is a configuration parameter. Therefore, the IDS can have a rule that
would monitor for normal levels of transmission power, and if an attacker attempts to jam the network, the IDS would raise a flag.

6.3. Data-Link Layer

In WirelessHART, the devices use designated time slots and frequency hopping for communication. Therefore, the IDS must monitor the nodes for adhering to the correct frequency sequence and time slots at all times. Otherwise, a compromised node can cause numerous collisions at the MAC layer by transmitting at a wrong frequency or in a wrong time slot. Therefore, there should be a mechanism by which the field IDS keeps track of transmission schedule of different nodes.

It is also possible that the attacker would change the sender field of a packet to match the time-slot it is sending the packet in. This strategy will make the attack harder to detect by the monitoring system. However, this requires the attacker to impersonate another node. This will be a difficult task to accomplish if the network deploys an access control list that maps the node ID to the MAC address and location of the node.

6.4. Routing

Given that availability is the most critical requirement of a SCADA system, the routing protocol needs to ensure that the packets sent by sensor nodes arrive at the master station consistently and at all times. To guarantee this availability, the WirelessHART standard suggests using full mesh routing. Fully redundant routing requires both spatial and temporal diversity. WirelessHART covers spatial diversity by enabling each node to discover multiple possible parent nodes and then establish links with two or more, and the temporal diversity is handled by retry and failover mechanisms.

Each node in the WirelessHART protocol implementation maintains its own neighbor list consisting of parent nodes and child nodes. A node may have as many parents as required, i.e., this is a configurable parameter. As mentioned earlier, the node message transmissions occur in one time slot and on one frequency (TDMA and frequency hopping). In each time slot a message is sent, and the sending node switches to receiving mode and waits for an acknowledgment (ACK). Should an ACK not arrive within the time slot, the sending node will retry in the next available slot. NACKs are generated for a number of reasons: invalid checksum (FCS), invalid message integrity code (MIC), or a full message queue by the receiving node. Each node in the WirelessHART keeps track of the missing ACKs and NACKs. If a number of transmissions have no acknowledgment, the sender node considers the path invalid and initiates communication with the next available node on its neighbor list.

6.5. Traffic Characteristics

To issue a live incident response, the IDS must collect and analyze volatile and nonvolatile network traffic data, in addition to the previously mentioned factors. For our purposes, traffic characteristics consist of two components: traffic load and traffic pattern. Traffic load is concerned with the amount of information being passed around in the network. In the WirelessHART specifications, for example, the maximum payload of the packet is 127 bytes. In addition to the packet size, the load is concerned with periodicity of the transmission of packets. The sensors send their monitored data periodically to the master (i.e., network manager), for example, every 30 seconds. Moreover, the 'physical health reports', which carry the statistical information of the wireless channel, are sent once every 15 minutes from each node. Therefore, the traffic load can be approximated fairly accurately. If the IDS observes a much higher/lower traffic load than expected, or if the packets are larger than 127 bytes, it could raise a flag for suspicious activity.

Traffic pattern refers to the communication pattern among the nodes in the network. For example, based on the discussion of the SCADA system layout, we can identify the following patterns: Master to slave, Network manager to field devices, HMI to masters, and Node-to-node communication at the network level, to route packets to the master. Therefore, the IDS can monitor the traffic for legitimate patterns. If at any point it observes unusual traffic, such as node to node at the sensing level (and not the network level), it can raise a flag or can monitor the activity of the node more closely.

6.6. Communication Datagram

The communication datagram specifications in WirelessHART give the details of the required fields in a communication packet. It specifies the format of each field, and the valid range of values for each field. WirelessHART datagram specifications include [3]: 1) Command priority: the IDS needs to monitor that the sensors do not send low-priority messages as high priority, 2) Data type: the data is
command, process data, alarm or normal, 3) Packet type: acknowledgement packets that include timing information to synchronize the TDMA (time-division multiple access) operations, and 4) Bandwidth request: this is used to provide elasticity of event traffic and ad hoc request/response maintenance messages. A generic communication packet format supported by WirelessHART specifications can be found in [3].

7. Policy Rules for Attack Detection

We outline our IDS policies, discuss which attacks they are targeted for, and which IDS (central or field) implements these policies. We assume that there is an access control list at the network manager, and the nodes have unique link keys, so that there is no need to define policies for the Sybil and cloning (replication) attacks.

**Policy rule 1:** Monitor packet count for each node in terms of the byte size of each packet and frequency of transmission. As pointed out in the previous section, the communication datagram is well defined with respect to the packet size and number of bytes. In addition, the slaves, i.e., sensor nodes, report to the master station in preconfigured intervals, for example every 30 seconds. This rule is applied at the field IDS and the central IDS and prevents the compromised nodes from sending too much or too little traffic. Especially if a compromised node tries to cause collision by sending a lot of traffic, the IDS is able to monitor the node’s activity level, and raise a flag if necessary.

**Policy rule 2:** Monitor physical health reports in order to determine if they are transmitted in the preconfigured intervals. For example, the current implementation has the nodes send a physical health report once every 15 minutes. This rule is implemented at the field as well as the central IDS, and helps prevent nodes from injecting too much (or too little) traffic into the network.

**Policy rule 3:** Monitor the request/reply traffic between the slave (sensor nodes) and the master station. The normal traffic pattern, as mentioned earlier, is a request from the master followed by the reply from the nodes, and in few cases a ’notification by exception’. Therefore, the IDS keeps track of how many requests/replies it sees in a given interval, which is a known number, and how many ’notification by exception’ packets. The request, reply and notification by exception packets are set with a flag in the packet header. If the requests are coming from any device other than the master (or in some cases a handheld device), then the IDS should raise a flag. Also, if a node sends too many exceptions, the IDS should notify the system engineer. This policy can be applied at both the field and central IDSs.

**Policy rule 4:** Monitor the individual datagram fields. The WirelessHART specifications give guidelines on the valid entries for each packet field. Therefore, the IDS must monitor the contents of each received packet to ensure valid entries for each item. For example, even though there is a field for sending control messages3 to the sensor nodes, it is not used by any currently deployed implementation. Therefore, the IDS should flag any packet that has the control option set. This rule must be implemented at the central IDS because the communication datagrams are encrypted, and the field IDS is not able to decrypt the contents of the packet in order to examine the packet fields. This rule protects against attacks in which the compromised node sends packets with invalid fields or contents.

**Policy rule 5:** Monitor the layer at which the communication among nodes is performed. As explained earlier, the communication among the sensor nodes is at the network layer in order to perform routing, and not at any other layer. This means that the sensors do not directly exchange information with each other, but rather use their neighbors for the purpose of routing their packets to the master. If a node attempts to communicate at any other layer with its neighbors, then the IDS should raise a flag to notify the system. This rule is implemented at the field IDS since the central IDS cannot monitor the local communication of the sensor nodes because of the communication range limits. However, the central IDS may correlate messages of this type from the field IDS components and infer the extent of unallowed communication.

**Policy rule 6:** Monitor the power level of the node communications. Since there are preconfigured levels of transmission power for the deployed nodes, the IDS can monitor for any deviation from the accepted levels. This rule is implemented at the field IDS and is designed to guard against a jamming attack. Again, the central IDS can use correlation to infer the extent of the jamming. Even though jamming is hard to defend against, it is useful for the network personnel to know when their system is under a jamming attack.

**Policy rule 7:** Monitor frequency/channel hopping sequence. Based on the WirelessHART protocol, the sensor nodes agree on the frequency hopping sequence for the time slots in which they communicate with each other. Therefore, the IDS needs to monitor the nodes to ensure that they follow the agreed-upon sequence and do not deviate, which will result in high number of packet collisions. This rule must be implemented at the field IDS level.

**Policy rule 8:** Monitor the MAC layer authentication at the field IDS. If a packet fails the authentication, it may be due to an attacker injecting bad packets into the network. Therefore, the IDS can signal a possible intrusion. Table 1 summarizes rules used for each layer and threat.

For the lying nodes, the solution is to keep a history for each node. Table 1 summarizes rules used for each layer and threat. For the lying nodes, the solution is to keep a history for all the reporting nodes at the network manager. If there is a spike in the reported values, look at the neighbors’ values,

---

3 Control in this context means using the sensors as actuators to adjust a valve for example.
Table 1. Policy rules for each layer and threat. NA refers to Not Applicable.

<table>
<thead>
<tr>
<th>Physical layer</th>
<th>Threat 1</th>
<th>Threat 2</th>
<th>Threat 3</th>
<th>Threat 4</th>
<th>Threat 5</th>
<th>Threat 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule 6</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>Data link layer</td>
<td>NA</td>
<td>Rule 5</td>
<td>Rule 1,2</td>
<td>Rule 7,8</td>
<td>Rule 8</td>
<td>NA</td>
</tr>
<tr>
<td>Network layer</td>
<td>Rule 1</td>
<td>Rule 3,8</td>
<td>Rule 1,2,3</td>
<td>Rule 2,4,5</td>
<td>Rule 4,5,8</td>
<td>Rule 3</td>
</tr>
</tbody>
</table>

and flag if it is an anomaly. The anomalies and node misbehavior can be determined locally by comparing the values reported by neighboring nodes. Adding the IDS into the network requires very little overhead. As discussed earlier, the health reports are configured to be sent by the sensor nodes in the current implementations. Therefore, the only overhead is adding the extra supernodes that could be used for the field IDSs.

8. Intrusion Response Mechanism

After detecting an intrusion or misbehavior, it is essential that the system takes appropriate action to stop the attack. Passive responses are typical in the IDS, such as logging of the information, and real-time notification. However, for the IDS to be effective, especially in the context of SCADA applications, we must have an active response system. Given the lack of physical security in wireless sensor networks, it is essential that the IDS has an effective active countermeasure. The type of intrusion response depends on the level of confidence in the intrusion, and the risk that the attack (or intrusion) carries. Defining the threat level of each attack is dependent on the type of the SCADA network application (e.g., oil and gas, electric power), and can be identified by a 'threat score'. Determining the threat score is beyond the scope of this paper. We assume that the risk analysis is done and the threat scores are loaded into the IDS by the network administrator. Based on the threat score and the confidence in the intrusion detection (e.g., with x% confidence, nodes X, Y, and Z are compromised), an appropriate measure can be taken, such as, removing the compromised nodes from the network. When the IDS suspects that an outsider has picked up the frequency hopping schedule of the nodes and is jamming a portion of the network, the central IDS can send a request to the nodes, possibly via the network manager, to change their frequency hopping sequence. This will prevent the attacker from continuing to jam the network by using the old frequency hopping sequence.

9. Conclusion and Future Work

Wireless sensor networks are becoming more prevalent in process control system applications, such as SCADA systems. Given the impact of attacks on the SCADA system on the nation and overall economy, it is crucial that the integration of sensor networks into the existing SCADA infrastructure does not increase the security threat of these systems. In this paper, we focused on the problem of intrusion detection, and proposed a multi-layer model-based intrusion detection system for SCADA applications. Our proposed solution is a combination of central and distributed IDS agents that work together to detect anomalous behavior among sensor nodes. We described normal behavior at various levels of the network stack. We are currently in the process of building a SCADA simulation testbed [6] using Simulink and other simulation packages. After the completion of the simulation testbed, we plan to implement our IDS policies on the SCADA testbed.

References