Understanding Total System Assurance: the Case of Mobile Agent-based Wireless Sensor Network Systems

Kassem Saleh, Anwar Al-Yatama and Nidal Nasser
Kuwait University

ABSTRACT

Total System Assurance (TSA) deals with assuring the security of all system components by considering all potential risks. This comprehensive approach to system assurance tackles security from multiple points of views, thus ensuring the highest possible level of assurance. In this paper we illustrate the TSA approach by considering a complex system running a mobile agent-based wireless sensor network (MA-WSN). Security in WSNs has been addressed extensively in the literature, however a comprehensive and integrative system security and assurance has not been considered. Total assurance in MA-WSN applications relies on the security of the mobile agents, and the mobile agent platform, in addition to the security of the wireless sensors and the application servers. Total assurance is tackled in a generic and comprehensive manner by considering a mixture of three approaches to addressing security concerns in systems. These approaches are based on the elicitation of security requirements, the misuse case-based threat model, and the prevention-detection-response model.

Keywords : assurance, security requirements, threat model

1 INTRODUCTION

Assurance and Security in complex systems are very important issues to tackle in a systematic and comprehensive approach. Knowing that reaching total security is an ideal goal of such approach, we argue that using a system's based comprehensive approach would bring system's assurance and security steps closer to perfection.

Educators in the field of assurance and security have the responsibility to disseminate the idea of total assurance and security. Based on the important principle stating that your system is as secure as your weakest point [1], failing in addressing all the possible weaknesses and vulnerabilities in your system lead to future and potential security breaches.

In this work, we consider mobile agent-based wireless sensor network applications to illustrate the need for total assurance when dealing with a complex system containing complex components and interfaces and providing possibly critical services at its public interfaces. Existing research in this area address only pieces of this complex system separately [2-7]. We feel there is a need to consider the whole system assurance and security in an integrated and comprehensive way. We propose to achieve this by considering our total system assurance (TSA) approach consisting of 1) the identification of system components and their interfaces, 2) the comprehensive elicitation of security requirements [8], 3) the identification of all potential threats to confidentiality, integrity, availability and accountability, 4) the identification of vulnerabilities that can be prevented or detected and dealt with should they be exploited. In this paper, due to the lack of space, we address the first three elements of the approach.

The rest of the paper is organized as follows. Section II identifies the system components and interfaces that must be assured in a MA-WSN system. Section III discusses the security requirements for MA-WSN system components and interfaces. Section IV discusses the various threats to security. Section V concludes the paper.

2 SYSTEM COMPONENTS AND INTERFACES

To develop a plan for a comprehensive and total assurance in a MA-WSN system, we need to identify the critical system components and the interfaces through which these components interact.

System components

A MA-WSN system is composed of five types of components or assets, namely, software, hardware, information, people, and procedures and policies. In our study, we concentrate on the first four types of system components.

To start the process of developing a comprehensive plan for system security, we need to identify the main components of the system we plan to secure. The main assets or components in a typical MA-WSN system are:

Hardware components: 1) sensor nodes (or sensors) that can be of different types: basic sensor nodes, aggregator or cluster heads, and base nodes. A typical sensor includes hardware components such as: the different specialized sensors, the microprocessor, the battery for power supply, the permanent and volatile memory and the communication processor, 2) One or more application servers to manage the application and collect and analyze information. 3) wired or wireless network devices connecting the wireless sensors to the application servers.

Software components: 1) software embedded in the sensor devices used to perform the necessary networking and processing functions. 2) mobile agent platform software residing at each sensor and responsible for receiving, launching and dispatching mobile agents. 3) software embedded inside the mobile agent either residing in the sensor or in transit between sensors. 4) software residing in the application servers to process and manage the collected information.

Information: 1) information collected and stored locally in the sensor's permanent memory or communicated to other sensors, 2) information carried inside the mobile agents themselves, 3) information saved inside the mobile agent platform, and finally, 4) information saved in the application server.

People: A (human) user will be able to use and manage the MA-WSN system by interacting with the application server.

Figure 1 shows the various components existing in a MA-WSN system.

System interfaces

To provide the services required from the MA-WSN system, its components must interact in an orderly and secure fashion. To be able to secure access at the various internal and external interactions points, all interfaces must be identified. Figure 2 below shows an interaction diagram that includes all the possible interactions and interaction points in a MA-WSN system. The possible interactions are listed below.

A sensor node interacts with other sensor nodes and with its own local platform. In addition, the sensor node may be required to interact with an agent in transit. Finally, the base (sensor) node will typically interact with the application server.

An agent interacts with the sending and receiving mobile agent platform. However, depending on the application, the agent may be required to interact with other co-located agents in transit or with the sensor node itself.

A platform interacts with the agents in transit and with the sensor node in which it is installed. Moreover, in some applications, a platform may be required to interact with other platforms and with the application server.

The application server interacts with the base sensor node and possibly with mobile agent platforms. In addition, the application server interacts with the human operator administering the system.

Figure 2 shows the various interfaces along which interactions among the MA-WSN system components interact. Access control restrictions governing the interactions are addressed in Section III.

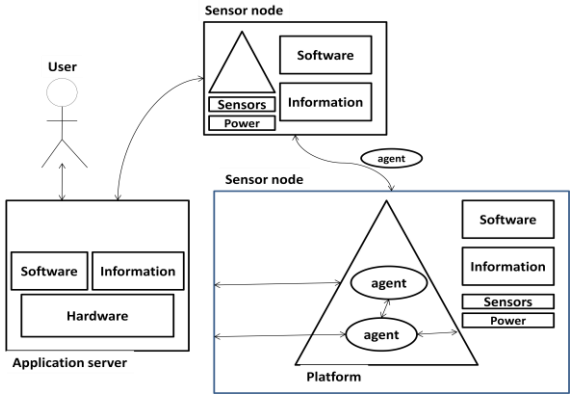

Figure 1. MA-WSN components.

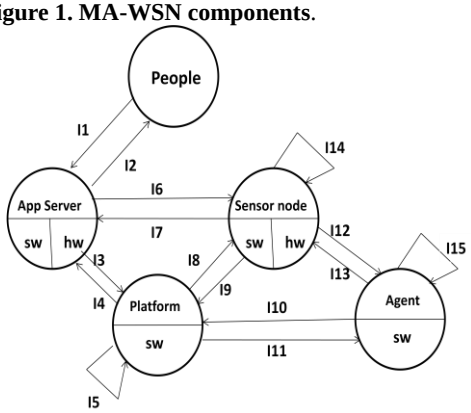

Figure 2. Interfaces in a MA-WSN system.

3 REQUIREMENTS-BASED VIEW

In this approach, security is addressed in a holistic and top-down manner. Starting from the expected services to be provided by the system to secure, security requirements are identified. These requirements are elicited using a refinement of the taxonomy of security requirements types introduced by Firesmith [8]. The types of security requirements that need to be elicited are: access control requirements including identification, authentication, and authorization requirements, privacy requirements, immunity requirements, integrity requirements, intrusion detection and prevention requirements, accountability requirements, auditing requirements, availability requirements, physical security requirements, system maintenance security requirements, and finally, the security standards conformity requirements. In the following we refer to the mobile agent-based WSN (MA-WSN) as simply the application.

Access control requirements

Access control can be achieved using three levels of layered controls, namely, identification, authentication and authorization. An access control requirement addresses the level at which a system component needs to identify, authenticate and authorize its interfaces before interacting with them. Depending on the application running on the system, different access control requirements will be needed for different types of components.

Typically, for example, a user access to the application server requires the user to identify herself over the people-server interface. The user's identity must be authenticated. Then based on the type of user (normal versus administrator) different authorization and access privileges will be assigned.

Table 1 can be used as a template to guide the capture of all relevant access control requirements in a MA-WSN system. For each of the system interfaces identified in Figure 2, we should determine the access control requirements to impose for the specific MA-WSN system. To identify the appropriate access control requirements, we can classify the identified interfaces according to the frequency of access to the interface and the criticality (or potential of risk damage) of the access. The combination of frequency and criticality can be used to determine the level of identification, authentication and authorization needed for that access. For example, for the people-server interface (I1) a stronger level of identification and authentication will be needed compared to the level needed for the sensor-agent interface (I12). Table 1 is filled with typical levels of frequency and criticality in MA-WSN system.

Notice that these requirements need not be symmetric or similar for both directions over the interface. For example, when the base sensor wants to communicate with the application server, it needs to be identified, authenticated and authorized by the server but not vice versa. Similarly, the user must be identified, authenticated and authorized when it communicates with the application server but not vice versa.

Integrity requirements

An integrity requirement specifies the extent to which the application must guarantee that the integrity of the assets is preserved. In MA-WSN applications, the main assets of interest are the information in transit and in storage, and the software inside the platform, the sensors, the server and the mobile agents. Tamper-proof information and software may be needed to guarantee integrity requirements.

Accountability requirements

An accountability requirement specifies the extent to which the application must guarantee that no participant in an interaction will be able to deny having committed an activity while interacting with other participants. These requirements are also referred to as non-repudiation requirements. For example, an administrator interfacing with the application server should not be able to deny performing any of its authorized functions. A mobile agent should not be able to deny having interacted with a sensor device, with a platform hosted inside a sensor device, or with another agent in transit at a platform.

Table 1. Access control requirements for system interfaces

Interface	Frequency	Criticality	Identification	Authentication	Authorization
I1: people-server	M	H	√	√	√
I2: server-people	M	L	√		
I3: server-platform	L	L	√		
I4: platform-server	L	H	√	√	√
I5: platform-platform	H	H	√	√	√
I6: server-sensor	H	L	√		
I7: sensor-server	H	H	√	√	√
I8: platform-sensor	H	M	√	√	√
I9: sensor-platform	H	H	√	√	√
I10: agent-platform	H	H	√	√	√
I11: platform-agent	H	L	√		
I12: sensor-agent	H	L	√		
I13: agent-sensor	L	M	√	√	√
I14: sensor-sensor	H	M	√	√	√
I15: agent-agent	L	M	√	√	√

Privacy requirements

A privacy requirement specifies the extent to which the application must guarantee that private information collected within the application or imported from other sources is protected against unauthorized accesses. For example, a privacy requirement states that sensitive and private information stored at the application server and in base sensor nodes are shielded or protected from unauthorized users, agents or sensors.

Immunity requirements

An immunity requirement specifies the extent to which the application must guarantee that its critical assets are immune from infected software or information. For example, an immunity requirement states that the application server should prevent the download of unauthorized software. Another requirement states that the mobile agent should protect itself from malicious code.

Intrusion detection and prevention requirements

An intrusion detection requirement specifies the extent to which the application is able to detect and prevent any hostile intrusion attempting to access any asset or to make unauthorized modifications to the system assets. For example, the sensor node is able to detect any attempt by a malicious mobile agent to reach the platform residing in the sensor. Another example is the ability of a mobile agent to detect and prevent a malicious agent from accessing or modifying its internal information or code. Clearly, access control mechanisms play a role in detecting and preventing intrusions.

Auditing requirements

An auditing requirement specifies the extent to which the application must collect security audit information at different levels of granularity. These requirements should be supporting the accountability or non-repudiation requirements. For example, each of the reported aggregated information at the base node should include the reporting sensor and the time of reporting. Similarly, each action performed by the administrator at the application server should be properly logged.

Availability or survivability requirements :

An availability requirement specifies the extent to which the application must be available when and after one of its assets was subject to a malicious attack or to a natural disaster. For example, an availability requirement states that the application must continue operating even if up to 50% of the participating sensors are inoperable or unreachable. Another requirement states that a mobile agent must survive an attack on the sensor it reaches.

Physical security requirements

A physical security requirement specifies the extent to which the system assets are physically protected from damage or harm caused by natural or human made incidents or disasters. For example, the application server must be physically protected against fire and flooding, and can only be reached by properly identified, authenticated and authorized personnel. Similarly, a sensor node must be camouflaged as much as possible without affecting the quality of communication with neighboring nodes.

System maintenance security requirements

A system maintenance security requirement specifies the extent to which the system must be secure during and after maintenance activities are performed. For example, the application must have the same or higher level of security during the maintenance window or after completion of the maintenance activities.

Security standards conformity requirements

A security standard conformity requirement specifies the types and levels of conformance of the application to the various professional, international, military, internal or country standards. For example, the organization hosting or administering the application must be ISO certified for quality and security.

Threat-based view

In this approach, security requirements are described by anti-requirements in terms of 'what are the security problems that may occur? And what the attacker or misuse can do?'. These anti-requirements can be based on misuse cases or abuse cases. What can go wrong to the confidentiality, integrity,

availability and accountability (CIAA)? What are the possible threats related to interception, interruption, insertion or fabrication, and alteration or modification of the various system elements (i.e., software, hardware and information).

A threat is the possibility of a bad thing occurring and affecting an asset or a resource. A vulnerability is a weakness that can make the threat a reality when exploited by an attacker. We say an attack enacts a threat to an asset by exploiting a vulnerability leading to that threat, and ultimately resulting in the deterioration of the quality of delivered services.

Having identified our main assets, we would like to identify all possible threats that the assets can be subjected to. To ensure that we capture these threats, we approach our threat identification process from different points of view leading to a comprehensive threat model. Threats are enacted by vulnerabilities that allow the attacker to launch interception, interruption, alteration or fabrication attacks. A successful interception attack compromises the confidentiality requirements, and a successful interruption attack compromises both availability and accountability requirements, whereas both alteration and fabrication attacks compromise integrity and accountability requirements. The different types of attacks leading to the compromise of the basic security requirements including confidentiality, integrity, availability, and accountability, as they relate to the different assets identified earlier, are discussed in the following.

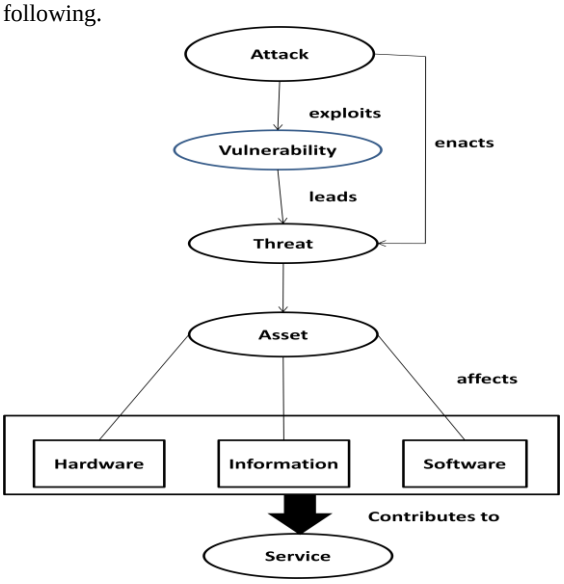

Figure 3. Threat model elements.

Interception involves a) the eavesdropping on communication links, b) the capturing of information inside computing and networking devices, and c) the capturing of the processing software. Interruption involves a) the removal, overloading or disconnection of a networking or computing device, and b) the deletion of information and software. Alteration or modification involves a) changes to the hardware configuration, b) changes to the processing software and c) changes to the information while in storage or in transit. Fabrication involves a) the insertion or addition of a hardware device, and b) the insertion or addition of information in both networking and computing devices, and c) the replay or injection of old information.

Threats to confidentiality

Confidentiality applies mainly to the information asset. Information can be in different states: stored in the sensor, in transit between nodes, in transit inside mobile agents traveling between nodes, or being processed by software at the sensor node, software inside the mobile agent or software inside the mobile agent platform. Confidentiality of information is compromised when a malicious authorized access or an unauthorized access to the information occurs. Exposing information to an attacker may lead to undesirable consequences, the severity of which depends on the value and criticality of the compromised information.

Possible attackers include: a malicious mobile agent (i.e., an injected mobile agent carrying malicious code), a genuine mobile agent carrying malicious (compromised) code, a malicious platform (i.e., an injected platform carrying malicious code), a genuine platform carrying malicious (or compromised) code, a malicious (injected) sensor node, or a genuine sensor node unknowingly carrying malicious code. For example, a malicious sensor may intercept information in transit or in a mobile agent to leak it to a malicious base node. A malicious mobile agent may also leak the information it carries to a malicious platform residing at a genuine or malicious sensor node. A malicious platform may forward a mobile agent or mobile agent information to a malicious sensor node. Finally, a malicious agent may interact with another genuine agent visiting the same sensor's platform.

Threat to confidentiality can also apply to software (as a special type of information) under certain scenarios. For example, a malicious agent may copy the software of another agent and leak it to a third party like another malicious agent, a malicious platform, or a malicious sensor. Knowing the logic inside decision-making software is potentially a serious threat to confidentiality. Agent software may encode the way the agent behaves and makes decisions. In this case, the agent software is considered to be valuable and confidential information threatened by an attacker. The same applies to the decision-making software inside the agent platform. Finally, exposing the decision-making software inside certain sensor nodes may also be considered a threat to

Threat to integrity

Integrity is mainly related to the proper modification of information by authorized users. However, integrity of software is also important in MA-WSN systems as we shall see in the following. Integrity of information or software is compromised by a malicious attacker in order to modify the information and affect the decision making process used to process this information.

Possible attackers include: a malicious agent modifying the information and software carried inside another genuine agent co-located at the same platform, a malicious agent modifying the information and software stored inside the platform residing in a sensor node, a malicious agent modifying the sensor's data and configuration information, a malicious platform modifying the information and code of a genuine agent when visiting that platform, a malicious attacker intercepting and modifying either or both the information or the software inside an agent in transit between sensors, and finally, a malicious sensor software modifying either or both of the information and the software residing in the agent or the platform itself. Modifying information leads to the transmission and processing of the wrong information to serve the attacker's goals. However, modifying the agent's or the platform's software leads to taking decisions that suit the attacker's goals, and pose a serious threat to the MA-WSN application and its provided services.

Threats to availability

Availability is mainly related to the proper offering of the services anytime they are requested by properly identified, authenticated and authorized (i.e., legitimate) users. By proper offering we mean that the quality of the provided services should be in conformance with that of the quality level initially agreed upon in a service level agreement. For the services to be available, all contributing assets must be available. The unavailability or the poor quality of the offered services can be due to the unavailability or poor quality of information, software or hardware. A malicious attacker may aim at rendering any of these critical assets unavailable or lowering the level of acceptable services they render.

Possible attackers include: (1) a malicious platform launching a denial of service attack by overloading another platform, a sensor device or an application server, (2) a malicious platform capturing or blocking a received mobile agent or destroying mobile agent information and software, (3) a malicious agent deleting the receiving platform or sensor information and software.

Threats to accountability

Accountability applies mainly to the saved, processed or transmitted audit information as a critical asset. Audit information stored in the agent, the platform, the application server, or the sensor device can be used to enforce accountability. Ideally, by analyzing this critical information, any malicious attack can be linked or traced back to its originator. The unavailability or the corruption of audit information would hinder the accountability enforcement goal of a security system and consequently degrade the overall security.

Possible attackers include: (1) a malicious platform modifying, inserting or deleting audit information stored inside a mobile agent or inside a received sensor, (2) a malicious agent modifying, inserting or deleting audit information stored inside a receiving platform, application server or a sensor.

Type of attack	Security requirement affected	Hardware			Software			Information			
		Sensor devices	Network devices	App. servers	Mobile agent sw	Platform software	Sensor software	Mobile agent information	Platform information	Sensor information	Server information
Interception	Confidentiality		√		√	√	√	√	√	√	√
Interruption	Availability	√	√	√	√	√	√	√	√	√	√
	Accountability										√
Alteration	Integrity				√	√	√	√	√	√	√
	Accountability							√	√	√	√
Fabrication	Integrity	√		√	√			√	√	√	√
	Accountability							√	√	√	√

4 CONCLUSIONS AND FUTURE WORK

Due to the increasing complexity of systems and the need for assuring and securing them, it is important to adopt and train assurance professionals on the use of an integrative and comprehensive approach to system security. In this work, we have illustrated our total system assurance approach using a complex system example involving mobile agents in a wireless sensor network application. Our approach considers and integrates various views of assurance including building a comprehensive requirement model, a threat model and an operational model based on the prevention, detection and response to threats and abuses. We intend to develop various artifacts to support our approach and formalize it further.

Table 2. Attacks, requirements and system components.

5 ACKNOWLEDGMENT

The authors would like to acknowledge the support of this research by a Kuwait University Research Grant.

6 REFERENCES

C. Pfleeger and S. Pfleeger, Security in Computing, Prentice Hall, 4th edition, 2006.

[2] J.P. Walters et al., Wireless sensor networks security: A Survey, Chapter 17, Security in Distributed, Grid and Prevasive Computing, CRC Press, 2006.

[3] W. Cockayne, and M. Zyda, (1998), *Mobile Agents*, Manning, Pearson Education, Harlow.

[4] Kiniry, J. and Zimmerman, D. (1997), ``A hands-on look at Java mobile agents'', *IEEE Internet Computing*, July, pp. 21-30.

[5] D. Lange and M. Oshima, Programming and Deploying Java Mobile Agents with Aglets, Addison-Wesley, 1998.

[6] M. Chen, et al., "Applications and design issues for mobile agents in wireless networks", IEEE Wireless Communications, December 2007, pp. 20-26.

[7] G. Karjoth et al., 'A security model for Aglets', *IEEE Internet Computing*, July-August 1997, 68–77.

[8] D. Firesmith, Engineering security requirements, *Journal of Object Technology Vol. 2, No. 1*, pp. 53-68, 2003.