Application of SemIoTica to the Development of a Prototype of an Intelligent System with IoT in Single-Family Aquaponics at the Tecno Academia Popayán

Oscar Santiago López Erazo[15] [0000-0002-8588-8365], Juliana Delle Ville[2] [0009-0007-7888-7544], Giuliana Maltempo[2] [0009-0005-7441-5828], Adriana Gómez[3] [0009-0008-5686-6408], Juan Camilo Ortega Erazo[4] [0000-0001-8322-1885], Luis Freddy Muñoz[5] [0000-0002-8172-0530], Julio Ariel Hurtado[1] [0000-0002-2508-096], Leandro Antonelli[2] [0000-0003-1388-0337]

[1] IDIS, Universidad del Cauca
[2] Lifia Facultad de Informática, Universidad Nacional de La Plata, Argentina
[3] Universidad Tecnológica de Pereira, Colombia
[4] SENA, Servicio Nacional de Aprendizaje
[5] Logiciel, Fundación Universitaria de Popayán

Abstract. The increasing adoption of Internet of Things (IoT) technologies in agriculture has led to improved decision-making processes. However, deploying IoT devices in agricultural systems raises concerns about security vulnerabilities and potential cyber threats. In this context, SemIoTica emerges as a method to systematically identify and mitigate security risks in IoT solutions for agriculture. This paper presents a case study that applies the methodology to identify the security vulnerabilities of a prototype aquaponics system, a tangible example of IoT application in smart agriculture. SemIoTica comprises four steps: (i) scenarios are described for the intended software, (ii) scenarios with incorrect uses of the system are described, (iii) these scenarios are translated into security scenarios using a set of rules, and (iv) the security scenarios are refined. Therefore, based on SemIoTica's four-step approach, correct and incorrect system use scenarios were systematically analyzed, and security scenarios were derived and refined to address the identified vulnerabilities. This paper presents an empirical application of the methodology in identifying and mitigating security risks in IoT agriculture, providing valuable information for developers and agriculture practitioners. Future approaches from this research may explore further refinement of security analysis methodologies and the development of robust cybersecurity measures tailored to the unique challenges of IoT applications in agriculture.

Keywords: IoT, Security Scenario, Smart Farming, Industry 4.0, IoT Requirements

1 Introduction

The quality and quantity of food have increased in recent years due to industrial growth and newer methods in the agricultural field [1]. Such technologies as the Internet of Things (IoT) have been introduced to promote better crop efficiency as well as to address the challenges faced by contemporary agriculture [2]–[4]. IoT is defined as a network of interconnected objects such as sensors, actuators, devices, and others, which

are grouped together for information exchange and collection of context-related data, with the aim of supporting decisions, enhancing processes and offering comprehensive solutions across the virtual and real worlds [3]–[5].

It is in this context that SemIoTica [6] acquires relevance, since it proposes a method for specifying security scenarios that integrate requirements and architecture elements into IoT agricultural solutions. The method comprises four steps: (i) scenarios for the intended software are described, (ii) scenarios with incorrect uses of the system are described, (iii) these scenarios are translated into security scenarios using a set of rules, and (iv) the security scenarios are improved. Furthermore, the authors present a prototype that uses the algorithm described in the proposal to help strengthen the incorrect use scenario based on the correct use scenarios; afterwards, the expert only needs to complete the information for the analysis and subsequent derivation of the security scenario.

Smart farming improves conventional agricultural practices by introducing technology into the field. This is beneficial; however, the deployment of devices might put agricultural activity at risk. Having devices connected to the internet can expose the system to vulnerabilities and attacks, exposing it at the level of security. Vulnerabilities, risks and threats to equipment and data in modern agriculture can be increased by harsh environmental conditions. Security in IoT has acquired utmost importance as there are more chances that hackers initiate an attack [7]. Another important factor is cybercrime or an unauthorized person accessing the system to cause harm or leak sensitive information [2], [8]. It is therefore crucial that the following items be taken into account when addressing security: (i) undesirable threats, (ii) protection of the surrounding context from damage, (iii) avoidance of physical damage, (iv) access control policies and authorization mechanisms [9]–[11].

As the previous items are relevant, vulnerabilities become important due to the risk factor associated with them in IoT agriculture systems. Systems like the single-family aquaponics system prototype use low-cost IoT technology [12] for the purpose of allowing users to remotely monitor key aspects of the system and provide an optimal environment for fish and plant growth. The prototype aims to develop a comprehensive IoT-based platform for collecting and analyzing vital data from aquaponic systems in real time. This is an IoT application in smart farming, as it supports the agricultural processes of aquaponics, defined as the union between aquaculture and hydroponics. More precisely, it consists of an agricultural method where fish waste is transformed into essential nutrients for plants. The main purpose of the system is to constantly monitor salinity, temperature, water level, pH, water quality, and sunlight, among others [4].

The SemIoTica approach and the algorithm-generated response can significantly enhance the misuse scenario by providing relevant additional information. This paper reports a preliminary evaluation of the application of the SemIoTica approach in a real project, namely the aquaponics project with the objective of identifying security scenarios for applications in IoT agriculture. This paper is organized as follows: Section 2 briefly describes the SemIoTica approach, and Section 3 describes related work. Next, Section 4 describes how SemIoTica is applied to the prototype. Section 5 presents a discussion. Finally, Section 6 presents conclusions and future work.

2 Background

This section describes SemIoTica, the method used to evaluate the security vulnerabilities in agricultural IoT applications. SemIoTica consists of 4 steps [6], summarized in Table 1. The following subsections describe each step-in detail.

Table 1. SemIoTica's approach summarized

→ Scenarios	1. Description of scenarios with correct use of the intended application	→	2. Description of scenarios with incorrect use of the intended application	→	3.Derivation of security scenarios	→	4. Refinement of the description of the security scenarios	→	Security Scenarios

2.1 Step 1: Description of scenarios with correct use of the intended application

This step entails the description of scenarios that focus on the correct use of the software application regarding security concerns. This step should be executed by a requirements engineer or analyst (or a group of them) who interacts with the experts of the domain (clients, users, and stakeholders in general) to capture the software application's requirements and specify scenarios. They should describe the functionality of the intended software, and they should also consider security concerns. Therefore, the analyst eliciting and defining scenarios should have some background in security non-functional requirements so as to consider this concern in the specification. The result step is a set of scenarios that describe the functionality.

2.2 Step 2: Description of scenarios with incorrect use of the intended application and algorithm

This step consists in analyzing the scenarios described in the previous step to find security issues. Issues that exploit the problems and compromise the security of the software application are described. Ideally, this step should be done by the same requirements engineer (or group of them) that participated in the previous tasks. The requirements engineer describes scenarios of incorrect use of the software application. Basically, they should describe scenarios that exploit possible vulnerabilities. This step is enhanced through the application of an algorithm. The operation of the algorithm can be summarized as follows: first, keywords related to specific attacks are identified in

existing scenarios (both correct and incorrect use scenarios). The catalogues are organized as tables, each heading corresponding to a column. Each row of the General Security Aspects catalogue contains information corresponding to one quality attribute (e.g., privacy, confidentiality, etc.). Each row of the Specific Attacks catalogue contains information describing one specific type of related attacks. The catalogues are then searched for the keywords in order to locate the row containing an occurrence of the specific attack. Once the relevant row is identified in both catalogues, the following information is extracted: affected security QA, attack involved, mitigation mechanism, and consequences in the agricultural industry. The algorithm concatenates the information extracted from the catalogues and appends it to a security scenario in a new field labelled 'threats'. The user can use this information to derive more robust and precise security scenarios

2.3 Step 3: Derivation of security scenarios

Is the derivation of security scenario. This step describes a set of rules to map the information contained in an incorrect use scenario so as to obtain a first draft of a scenario describing security concerns. It is worth mentioning that the incorrect use scenario will not provide enough information for a complete security scenario. The rules proposed here use only four attributes from the incorrect use scenario (title, context, actors, and resources) to fill out four attributes of the security scenario (stimulus, environment, source of the stimulus, and artifact). With this information, the following step is to refine the security scenario. In this step a draft scenario is created by filling in some of the fields in a security scenario (Table 2).

Table 2. Mapping rules used in the derivation of security scenarios

Attribute of the incorrect usage scenario	Attribute of the security scenario
Title	Stimulus
Context	Environment + Source of the stimulus
Actors	Sources of stimulus
Resources	Artifact

2.4 Step 4: Refinement of the description of the security scenarios

Some adjustments and improvements should be made to the scenarios derived from the mapping in the previous step. Some new information should be added, and some information should be rephrased. The requirements engineer should use their experience and knowledge to provide further information and paraphrase other based on the elicitation meeting and their expertise in the field as following: (i) an identification must be provided; (ii) the stimulus must be paraphrased; (iii) the environment and source of stimulus must be split into two attributes; (iv) the source of stimulus must be rephrased; (v) the artifact must be rephrased; and finally (vi) the response and the response measure must be completed from scratch. Although the mapping rules do not provide infor-

mation corresponding to these attributes, the information found in the rest of the scenario provides the necessary context so that the requirements engineers can describe these two last attributes. The requirements engineer should bear in mind that the response measure attribute, in particular, should be described with quantitative measures. The draft is refined on the basis of an expert's expertise. The draft is thus transformed into a security scenario.

3 Related work

Dorairaju [13] focuses on addressing security vulnerabilities in an IoT remote insect pest monitoring system, which is a critical component for smart agriculture. The objective of their work is to study the impact that digital data produces on the security aspect for an agricultural platform. Their case study investigates security challenges, sensor issues, and IoT devices. Using a case study involving a combination of investigations of IoT modules used in prototype field experiments and a review of related literature, a concrete guide is developed. Their most important contribution is a checklist for reviewing the security requirements of IoT. They also conclude that the people involved in these projects must have training in, and sufficient knowledge of, the principles of cybersecurity. Likewise, Riaz et al. [14] describe an evaluation framework for smart agricultural environments which contains execution scenarios of an agricultural environment with devices and sensors. Storylines are derived from these scenarios by providing real-time environments to expand and incorporate adaptive security frameworks and scenarios in IoT-based agriculture. As future work, they project to create a simulation of the generated scenarios.

A number of studies have considered mitigation for developers with little experience in IoT security [13], [15]–[17]. It is concluded that requirements engineers as well as software architects require specific and concrete methodologies to identify, understand and limit the risks associated with user privacy/security posed by IoT devices as well as threat scenarios and mitigation efforts. Rutledge and Massey [16] carry out an exploratory case study of the privacy policies of an IoT device (SmartTV), with the objective of characterizing the privacy protections and vulnerabilities associated with this class of devices. Among the most important results, they find that there is a greater risk of suffering damage to privacy due to SmartTVs, and that the user's privacy is further compromised by requiring a connection with the manufacturer background servers.

The work carried out by Demestichas et al. [8] also focuses on the vulnerabilities and threats of IoT agriculture systems. The authors carry out a review of the literature on the use of this technology in the aforementioned domain.

The article by Sicari et al. [18] proposes the use of Node-RED, a flow-supported programming tool specialized in IoT, along with a series of case studies related to different IoT contexts, one of which pertains to Smart Agriculture (an automated greenhouse management application) with the objective of collecting data on soil conditions and consulting a database. data that contains useful information about the types inside the greenhouse, the data is analyzed by said system, which decides whether to activate one or more actuators of the greenhouse. The authors find that IoT applications suffer

from four general problems: (i) scalability, (ii) inter-operability, (iii) security and (iv) privacy. With respect to security and privacy, it is mentioned that complex mechanisms must be integrated into the items mentioned above, policy management systems and tests in order to verify the viability of the proposed approaches [18], [19].

Finally, Wangyal et al. [20] identify risk factors on commercial operations using the risk breakdown structure method, with the objective of expanding the traditional view by including other risks such as physical risks. The authors conclude that applying IoT in different domains brings new challenges such as limitations in resources (energy, memory and computing capacity), and that security remains a high priority aspect for users. This supports the claim by Rettore [21] that conventional protection schemes used in traditional Internet or IoT may not be useful for agricultural systems, a fact that creates additional research opportunities.

4 Method

This paper adopted the method proposed by Wang and Xian [22] to carry out the case study, with the following proposed phases. The necessary resources are shown on Table 3. The rest of the section is organized in the following subsections: (i) Experimental procedures, (ii) Evaluation of the prototype using SemIoTica and the proposed algorithm and (iii) Report of findings (Conceptual Results) and acquired knowledge.

Table 3. Resources needed to carry out the case study

Element	Description
Research Question	Will SemIoTica help to improve experts' ability to identify security scenarios for an Agriculture IoT Prototype?
Instruments	(i) Prototype of an Intelligent System with IoT for the Monitoring of Variables in Single-Family Aquaponics at the Tecnoacademia Popayán: this research project focuses on citizen science through the development of a single-family aquaponics system, using IoT technology that allows users to remotely monitor key aspects of the system, providing an optimal environment for fish and plant growths (ii) A web implementation of SemIoTica called "Requirements Healer": a web application specialized in requirements management and requirements processing.
Participants	Four experts in IoT Security are described below (gender, age, culture, experience) : (i) Male, 30 ,Colombian, Software architect, (ii) Female, 25, Argentina, requirements expert, (iii) Female, 26, Argentina, requirements expert and (iv) Female, 32, Colombia, requirements expert.

4.1 Experimental procedures

The prototype called "Desarrollo de un Prototipo de Sistema Inteligente con IoT para el Monitoreo de Variables en Acuaponía Unifamiliar de la Tecnoacademia Popayán" [12] was chosen on the grounds that it applies IoT technologies to agriculture, thus enabling the application of SemIoTica.

To address this section, the definition of the protocol and procedures to follow to execute SemIoTica is carried out. The activities carried out in this phase are the following:

1. The prototype creator explains its design, concepts and operation through an online meeting.
2. Participants take a reasonable amount of time to understand the design and operation of the prototype.
3. The prototype is evaluated with SemIoTica by following the proposal's steps of and algorithm.
4. Findings, acquired knowledge, and results are reported.
5. The results are discussed from a conceptual viewpoint.

The details of phases 3, 4, 5 and 6 are reported below.

4.2 Evaluation of the prototype using SemIoTica and the proposed algorithm

The results of applying the SemIoTica approach to the aquaponics prototype are shown below. Figures 1-3 show the correct usage scenarios generated in step 1. Figures 4-6 show the corresponding incorrect usage scenarios generated in step 2.

Element	Description
Scenario Name	Oxygen Sensor
Goal	Measure oxygen in the tank
Context	A tank full of live fish
Resources	Oxygen
Actors	The sensor
Episodes	The sensor takes a measurement of oxygen

Fig 1. Correct use scenario for oxygen sensor

Element	Description
Scenario Name	Temperature Sensor
Goal	Measure the tank temperature to ensure the fish
Context	A tank full of live fish
Resources	Temperature
Actors	The sensor
Episodes	The sensor takes a measurement of temperature

Fig 2. Correct use scenario for temp sensor

Element	Description
Scenario Name	Acidity sensor
Goal	Measure water pH and ensure fish
Context	A tank full of live fish
Resources	Acidity (pH)
Actors	The sensor
Episodes	The sensor takes a measurement of pH

Fig 3. Correct use scenario for PH sensor

Element	Description
Scenario Name	Intervened/defective oxygen sensor
Goal	Measure oxygen in the tank
Context	A tank full of live fish
Resources	Oxygen
Actors	The sensor
Episodes	The sensor sends a signal to stop the oxygen pump. Fish die from asphyxiation

Fig 4. Incorrect use scenario for oxygen sensor

Element	Description
Scenario Name	Intervened/defective temperature sensor
Goal	Measure the tank temperature to ensure the fish
Context	A tank full of live fish
Resources	Temperature
Actors	The sensor
Episodes	The sensor sends a warning notification to the dashboard that the water is too cold/warm.

Fig 5. Incorrect use scenario for temp sensor

Element	Description
Scenario Name	Defective/interfered acidity sensor
Goal	Measure water pH and ensure fish
Context	A tank full of live fish
Resources	Acidity (pH)
Actors	The sensor
Episodes	The sensor sends a warning notification to the dashboard that the water has a low/high pH.

Fig 6. Incorrect use scenario for PH sensor

The output of the algorithm proposed in SemIoTica, which provides information on identified vulnerabilities, can be seen in Table 4.

Table 4. Result provided by the algorithm proposed in SemIoTica

Element	Description
Affected QA	Privacy, Confidentiality, Authenticity
Affected Layer	Perception Layer, Physical Layer
Layer Details	The first layer is composed of smart IoT sensing devices e.g., smart phones, RFID tags, sensors and actuators, etc. These components are able to automatically sense, collect and measure the various physical parameters e.g., temperature, humidity, location etc. Devices can store collected information inside themselves and sensors can store information into predefined sensor hubs (e.g., a microcontroller unit) to process them. The major functionalities of this layer are data sensing and data acquisition. Standardized plug-and-play mechanisms can be used with the various sensing devices. Furthermore, considering the scale of the number of things in an IoT system, sensing devices may be deployed simultaneously or over time according to the environmental context and practical requirements
Mitigation mechanism:	Two-Factor Authentication. RBAC ensures that only authorized users are given access to certain data or resources. It also supports three well-known security principles: information hiding, least-privilege, and separation of duties.
Impact	The collection of information regarding the type and possible usage of devices concerning agriculture projects. These security leaks can be used in order to get access to infrastructure and production standards as well as getting privacy data and compromising the privacy of the system. Theft and vandalism

The output shown in Table 4 will help strengthen the incorrect use scenario that is based on a correct use scenario. After analysis of the output, the expert completes the incorrect use scenarios with additional security information, as shown in Table 5.

Table 5. Incorrect use scenario for oxygen sensor strengthened with information provided by the algorithm

Element	Description
Scenario Name	Intervened/defective oxygen sensor
Goal	Measure oxygen in the tank
Context	A tank full of live fish
Resources	Oxygen
Actors	The sensor
Episodes	The sensor sends a signal to stop the oxygen pump. Fish die from asphyxiation

Threats	The QAs affected are Privacy, Confidentiality, Authenticity. The Layers affected are Perception Layer, Physical Layer. The major functionalities of these layers are data sensing and data acquisition. The suggested mitigation mechanism to correct the affected QAs is Two-Factor Authentication. RBAC ensures that only authorized users are given access to certain data or resources. The impact of not using this mechanism is that these security leaks can be used in order to get access to infrastructure and production standards.

Step 3 consists in deriving the draft security scenarios by mapping incorrect scenario fields to security scenario fields as follows: (i) title to stimulus, (ii) context to Environment + Source of stimulus, (iii) Actors to Sources of stimulus, and (iv) resources to artifact. The resulting scenarios are shown in Tables 6-8.

Table 6. Security scenario draft for oxygen sensor

Element	Description
Stimulus	Intervened/defective oxygen sensor
Environment + Source of stimulus	The sensor sends a signal to stop the oxygen pump. Fish die from asphyxiation: A tank full of live fish may be compromised because the Layers affected are Perception Layer, Physical Layer. The major functionalities of these layers are data sensing and data acquisition. The QAs affected are Privacy, Confidentiality, Authenticity. These security leaks can be used in order to get access to infrastructure and production standards. The suggested mitigation mechanism to correct the affected QAs is Two-Factor Authentication
Sources of stimulus	The sensor
Artifact	Oxygen

Table 7. Security scenario draft for temperature sensor

Element	Description
Stimulus	Intervened/defective temperature sensor
Environment + Source of stimulus	The sensor sends a warning notification to the dashboard that the water is too cold/warm: A tank full of live fish may be compromised because the Layers affected are Perception Layer, Physical Layer. The major functionalities of these layers are data sensing and data acquisition. The QAs affected are Privacy, Confidentiality, Authenticity. These security leaks can be used in order to get access to infrastructure and production standards. The suggested mitigation mechanism to correct the affected QAs is Two-Factor Authentication
Sources of stimulus	The sensor
Artifact	Temperature

Table 8. Security scenario draft for acidity sensor

Element	Description
Stimulus	Intervened/defective acidity sensor
Environment + Source of stimulus	The sensor sends a warning notification to the dashboard that the water has a low/high pH: A tank full of live fish may be compromised because the Layers affected are Perception Layer, Physical Layer. The major functionalities of these layers are data sensing and data acquisition. The QA´s affected are Privacy, Confidenciality, Authencity. These security leaks can be used in order to get access to infrastructure and production standards. The suggested mitigation mechanism to correct the affected QAs is Two-Factor Authentication
Sources of stimulus	The sensor
Artifact	Acidity (pH)

Finally, on Step 4 some adjustments and improvements should be made to the scenarios derived from the mapping in the previous step. Some new information should be added, and some information should be rephrased. In particular, the following points should be observed:

1. Identification must be provided
2. The stimulus must be rephrased
3. The environment and the source of stimulus must be split in two attributes
4. The source of stimulus must be rephrased
5. The artifact must be rephrased
6. The response and the response measure must be added

The final results, i.e., the complete security scenarios, are shown in Tables 9-11.

Table 9. Security scenario for oxygen sensor

Element	Description
ID	S-01
Stimulus	Oxygen sensor malfunctioning because it was Intervened/defective
Environment	Fish die from asphyxiation: A tank full of live fish may be compromised because the Layers affected are Perception Layer, Physical Layer. The major functionalities of these layers are data sensing and data acquisition. The QA´s affected are Privacy, Confidentiality, Authenticity. These security leaks can be used in order to get access to infrastructure and production standards. The suggested mitigation mechanism to correct the affected QAs is Two-Factor Authentication

Sources of stimulus	The oxygen sensor of selected prototype makes a measurement
Artifact	Oxygen
Response	The sensor sends a signal to stop the oxygen pump
Response Measure	Ensure the security of the system, it's important that attacks are detected quickly, ideally within 0,5 seconds.

Table 10. Security scenario for temperature sensor

Element	Description
ID	S-02
Stimulus	Temperature sensor malfunctioning because it was Intervened/defective
Environment	The water is too cold/warm: A tank full of live fish may be compromised because the Layers affected are Perception Layer, Physical Layer. The major functionalities of these layers are data sensing and data acquisition. The QAs affected are Privacy, Confidentiality, Authenticity. These security leaks can be used in order to get access to infrastructure and production standards. The suggested mitigation mechanism to correct the affected QAs is Two-Factor Authentication
Sources of stimulus	The temperature sensor of selected prototype makes a measurement
Artifact	Temperature
Response	The sensor sends a warning notification to the dashboard that the water is too cold/warm
Response Measure	Ensure the security of the system, it's important that attacks are detected quickly, ideally within 0,5 seconds.

Table 11. Security scenario for acidity sensor

Element	Description
ID	S-03
Stimulus	Acidity sensor malfunctioning because it was Intervened/defective.
Environment	The water has a low/high pH: A tank full of live fish may be compromised because the Layers affected are Perception Layer, Physical Layer. The major functionalities of these layers are data sensing and data acquisition. The QAs affected are Privacy, Confidentiality, Authenticity. These security leaks can be used in order to get access to infrastructure and production standards. The suggested mitigation mechanism to correct the affected QAs is Two-Factor Authentication
Sources of stimulus	The acidity sensor of selected prototype makes a measurement
Artifact	Acidity (pH)
Response	The sensor sends a warning notification to the dashboard that the water has a low/high pH

Response Measure	Ensure the security of the system, it's important that attacks are detected quickly, ideally within 0,5 seconds.

4.3 Report of findings (Conceptual Results) and acquired knowledge

The results found by participants when using the SemIoTica approach are:

1. According to the reported findings, it is found that there is a high possibility of frequently using the SemIoTica proposal because it contributes to the development of security scenarios for smart farming.
2. The proposal was easy to use and did not cause any setbacks because it is not necessary to have an expert to guide its use since the sequence of steps is clear and simple, making it easy for people to quickly use this proposal.
3. When using SemIoTica, it is observed that the stated steps are well defined, concrete and concise, which makes the proposal provide certainty to anyone who is applying it.
4. The information to create the scenarios and associate it with the algorithm must be in English.
5. The response generated by the algorithm was of great help to complement the misuse scenario by associating more information and contributing to improving it.
6. Following the steps of the SemIoTica approach and the algorithm contributes to improving the scenarios, enriching their information and helping the expert in their task of generating security scenarios in IoT Agriculture.
7. The resulting security scenarios have well marked security problems that affect the sensor.
8. The algorithm shows consistency because the information entered in the misuse scenarios of the three sensors have the same associated results.

5 Discussion

The results of SemIoTica's algorithm reflect what kind of vulnerabilities a system may have. In this case, this aquaponics project uses sensors and pumps to maintain the fish—the core of the system. Since the system's well-functioning needs periodical measurements, the sensors are a vital component, and they constitute the physical (or perception) layer. The physical layer plays an important role as it gathers information from the surrounding context in order to act upon said information. This information enables the normal unfolding of correct system behaviors.

Damage to the sensors can result from normal usage over time or due to environmental conditions like erosion. Hence durability must be considered an important variable in the system. Battery durability is another matter of concern as an excess of security algorithms can hasten battery depletion, depriving the system of the necessary equipment to maintain the fish. To extend trust in the sensors, actuators and other devices that make up an IoT system, it is necessary to improve the durability of materials

to prevent wrong information, false positives or unwanted behaviors within the application.

Adopting such technology as IoT in agriculture can be advantageous in terms of crop efficiency and support in the crop supply chain. However, the use of this technology can bring some problems because sensors, actuators and other devices are deployed that are connected to the Internet and that can be easily manipulated by third parties in order to cause damage. As the sensors are vital to maintain the fish, some malicious activities are attracted to this equipment. To harm the system, an attacker can intercept the measurements and modify it or store it because it contains sensitive information about the state of the core of the system, the fishes. Another kind of attack can be directly focused on taking the sensors out by accessing them and taking control or attacking them and depleting the energy or faking them. If the fish do not have the right conditions, they can die and therefore the system can no longer function.

It is necessary to keep in mind that if we use IoT in Smart Farming and Aquaponics, we must be extremely careful because the life of a living being such as a fish is involved. Therefore, it is important to mitigate any attempted damage to applications that use this approach, emphasizing the techniques suggested by the literature for IoT security. In order to mitigate those attacks, it's logical to have a two-factor authentication system, because it is difficult for the attackers to get into the device and finally to encrypt the measurements of the sensor to secure the sensitive information and have a hash to ensure the information is in a correct state.

The SemIoTica approach was very useful because it contributes to the security measurements and methods to protect the vital and important equipment and maintain secure the fish. The results are very promising and show effectively what are the vulnerabilities of this real case. In every scenario, it shows what layers are affected, what QAs are involved, what kind of attacks the system is vulnerable to and the countermeasures to deal with them. it contributes to the security measurements and methods to protect the vital and important equipment and maintain secure the fish.

6 Conclusions and future work

In this paper we dealt with an application of SemIoTica methodology in a real case of study. We applied a detailed analysis of the case study. Then we applied the SemIoTica approach in order to obtain the security scenarios. And finally, we discussed the results and made a reflection about it. The results obtained are very promising and show how effective is the SemIoTica method. In the resulted scenarios, shows what layers are affected, what QAs are involved, what kind of attacks the system is vulnerable to and the countermeasures to deal with them. Also the application of the algorithm show consistency in every scenario and highlighted the impact if any of this sensor stop working in the system

SemIoTica helps improve the ability of experts to identify security scenarios of an agricultural IoT prototype because: (i) it helps to build security scenarios in an agile, concise and precise manner with the steps involved given above, (ii) enrich them with

information provided by the proposed algorithm prototype and (iii) identify security scenarios effectively in intensive applications in IoT agriculture.

It is intended to continue applying SemIoTica in different agricultural IoT prototypes for the purpose of finding more results that lead to the improvement of the proposal, the algorithm, the designed and identified scenarios. As for the future work a detailed case study is planned about this method. A user perception survey on the prototype is also needed to continue enhancing the method.

References

1. B. B. Sinha and R. Dhanalakshmi, "Recent advancements and challenges of Internet of Things in smart agriculture: A survey," *Futur. Gener. Comput. Syst.*, vol. 126, pp. 169–184, 2022, doi: 10.1016/j.future.2021.08.006.
2. A. Yazdinejad *et al.*, "A review on security of smart farming and precision agriculture: Security aspects, attacks, threats and countermeasures," *Appl. Sci.*, vol. 11, no. 16, 2021, doi: 10.3390/app11167518.
3. S. Sotoudeh, S. Hashemi, and H. G. Garakani, "Security Framework of IoT-Based Smart Home," *2020 10th Int. Symp. Telecommun. Smart Commun. a Better Life, IST 2020*, pp. 251–256, 2020, doi: 10.1109/IST50524.2020.9345886.
4. V. K. Quy *et al.*, "IoT-Enabled Smart Agriculture: Architecture, Applications, and Challenges," *Appl. Sci.*, vol. 12, no. 7, 2022, doi: 10.3390/app12073396.
5. K. Ojo-Gonzalez and B. Bonilla-Morales, "Requerimientos no funcionales para sistemas basados en el Internet de las cosas (IoT): Una revisión," *I+D Tecnológico*, vol. 17, no. 2, 2021, doi: 10.33412/idt.v17.2.3303.
6. J. Hurtado *et al.*, "Semiotica : An Approach to Model Security Scenarios for IoT-Based Agriculture Software/SemIoTica : Un enfoque para modelar escenarios de seguridad para software de agricultura basado en IoT," doi: https://doi.org/10.22430/22565337.2923.
7. R. Yugha and S. Chithra, "A survey on technologies and security protocols: Reference for future generation IoT," *J. Netw. Comput. Appl.*, vol. 169, p. 102763, 2020, doi: 10.1016/j.jnca.2020.102763.
8. K. Demestichas, N. Peppes, and T. Alexakis, "Survey on Security Threats in Agricultural IoT and Smart Farming," *sensors*, 2020, doi: 10.1049/et.2012.0601.
9. S. El-Gendy and M. A. Azer, "Security Framework for Internet of Things (IoT)," *Proc. ICCES 2020 - 2020 15th Int. Conf. Comput. Eng. Syst.*, no. December 2020, 2020, doi: 10.1109/ICCES51560.2020.9334589.
10. A. Rizzardi, S. Sicari, and A. Coen-Porisini, "Towards rapid modeling and prototyping of indoor and outdoor monitoring applications," *Sustain. Comput. Informatics Syst.*, vol. 41, no. November 2023, p. 100951, 2024, doi: 10.1016/j.suscom.2023.100951.
11. J. Liu, Y. Xiao, and C. L. P. Chen, "Authentication and access control in the Internet of things," *Proc. - 32nd IEEE Int. Conf. Distrib. Comput. Syst. Work. ICDCSW 2012*, pp. 588–592, 2012, doi: 10.1109/ICDCSW.2012.23.
12. J. C. Ortega Erazo, "Desarrollo de un Prototipo de Sistema Inteligente con IoT para el Monitoreo de Variables en Acuaponía Unifamiliar de la Tecnoacademia Popayán." p.

1, [Online]. Available: https://cutt.ly/IoT-TecnoacademiaPopayan.
13. G. Dorairaju, “Cyber Security in Modern Agriculture. Case Study: IoT-based Insect Pest Trap System,” no. April, p. 81, 2021, [Online]. Available: http://www.theseus.fi/handle/10024/497436%0Ahttps://www.theseus.fi/bitstream/handle/10024/497436/Thesis_Dorairaju_Ganeas.pdf?sequence=2&isAllowed=y.
14. A. R. Riaz, S. M. M. Gilani, S. Naseer, S. Alshmrany, M. Shafiq, and J. G. Choi, “Applying Adaptive Security Techniques for Risk Analysis of Internet of Things (IoT)-Based Smart Agriculture,” *Sustain.*, vol. 14, no. 17, 2022, doi: 10.3390/su141710964.
15. D. Mauro, L. Melo, H. Lu, M. Damorim, and A. Prakash, “A study of vulnerability analysis of popular smart devices through their companion apps,” *Proc. - 2019 IEEE Symp. Secur. Priv. Work. SPW 2019*, pp. 181–186, 2019, doi: 10.1109/SPW.2019.00042.
16. R. L. Rutledge, A. K. Massey, and A. I. Anton, “Privacy impacts of IoT devices: A SmartTV case study,” *Proc. - 2016 IEEE 24th Int. Requir. Eng. Conf. Work. REW 2016*, pp. 261–270, 2017, doi: 10.1109/REW.2016.40.
17. M. Ye, N. Jiang, H. Yang, and Q. Yan, “Security analysis of Internet-of-Things: A case study of August smart lock,” *2017 IEEE Conf. Comput. Commun. Work. INFOCOM WKSHPS 2017*, pp. 499–504, 2017, doi: 10.1109/INFCOMW.2017.8116427.
18. S. Sicari, A. Rizzardi, and A. Coen-Porisini, “How to evaluate an Internet of Things system: Models, case studies, and real developments,” *Softw. - Pract. Exp.*, vol. 49, no. 11, pp. 1663–1685, 2019, doi: 10.1002/spe.2740.
19. M. Miki, T. Yamauchi, and S. Kobayashi, “Evaluation of Effectiveness of MAC Systems Based on LSM for Protecting IoT Devices,” *Proc. - 2023 11th Int. Symp. Comput. Networking, CANDAR 2023*, pp. 161–167, 2023, doi: 10.1109/CANDAR60563.2023.00029.
20. S. Wangyal, T. Dechen, S. Tanimoto, H. Sato, and A. Kanai, “A Study of Multi-viewpoint Risk Assessment of Internet of Things (IoT),” *Proc. - 2020 9th Int. Congr. Adv. Appl. Informatics, IIAI-AAI 2020*, pp. 639–644, 2020, doi: 10.1109/IIAI-AAI50415.2020.00131.
21. A. Rettore de Araujo Zanella, E. da Silva, and L. C. Pessoa Albini, “Security challenges to smart agriculture: Current state, key issues, and future directions,” *Array*, vol. 8, no. December, p. 100048, 2020, doi: 10.1016/j.array.2020.100048.
22. S. Wang and Y. Xian, “A case study on the efficacy of error correction practice by using the automated writing evaluation system WRM 2.0 on Chinese college students’ English writing,” *Proc. - 2011 Int. Conf. Comput. Inf. Sci. ICCIS 2011*, pp. 988–991, 2011, doi: 10.1109/ICCIS.2011.21.