Internet of Things 000 (2019) 100056

Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.elsevier.com/locate/iot

Research article

# Toward the automation of threat modeling and risk assessment in IoT systems

Valentina Casola[a], Alessandra De Benedictis[a,*], Massimiliano Rak[b], Umberto Villano[c]

[a] Department of Electrical Engineering and Information Technologies, University of Naples, Via Claudio 21, 80125, Italy
[b] Department of Computer Engineering, University of Campania Luigi Vanvitelli, Via Roma 29, 81031, Aversa, Italy
[c] Department of Engineering, University of Sannio, Via Traiano 3, 82100, Benevento, Italy

ARTICLE INFO

*Article history:*
Received 30 January 2019
Revised 3 May 2019
Accepted 4 May 2019
Available online xxx

*Keywords:*
IoT automated threat modeling
IoT automated risk assessment
IoT secure design

ABSTRACT

The Internet of Things (IoT) has recently become one of the most relevant emerging technologies in the IT landscape. IoT systems are characterized by the high heterogeneity of involved architectural components (e.g., device platforms, services, networks, architectures) and involve a multiplicity of application domains. In the IoT scenario, the identification of specific security requirements and the security design are very complex and expensive tasks, since they heavily depend on the configuration deployment actually in place and require security experts. In order to overcome these issues, we propose an approach aimed at supporting the security analysis of an IoT system by means of an *almost completely automated process for threat modeling and risk assessment*, which also helps identify the security controls to implement in order to mitigate existing security risks. We demonstrate the effectiveness of the approach by discussing its application to a home automation system, built on top of commercial IoT products.

© 2019 Elsevier B.V. All rights reserved.

## 1. Introduction

The Internet of Things (IoT) represents one of the most dynamic areas in the current IT landscape, due to the integration of highly heterogeneous technologies, and to the continuous rise of new IoT-powered applications in different domains, including smart cities/buildings, e-health systems, smart manufacturing systems and intelligent transportation systems. A side effect of such dynamism is the lack of a shared, commonly recognized and adopted IoT architectural and functional vision: the main vendors typically propose their own architecture, and the same happens in the research and academic fields [1]. However, some initiatives exist towards the standardization of the adopted vocabulary, the reference architecture and the design best practices (e.g., the ISO/IEC 30141 – Internet of Things Reference Architecture (IoT RA) standard [2]).

Due to the involved heterogeneity and to the resulting integration issues, designing secure IoT systems is very challenging and requires a deep knowledge of the adopted technologies and of the deployment configuration, including the specific installed devices, the network technologies used for their communication (ranging from ad-hoc, low-power connections to WiFi networks), the devised user interactions, and the adopted back-end technologies (e.g., cloud or on-premises services).

* Corresponding author.
*E-mail addresses:* casolav@unina.it (V. Casola), alessandra.debenedictis@unina.it (A. De Benedictis), massimiliano.rak@unicampania.it (M. Rak), villano@unisannio.it (U. Villano).

https://doi.org/10.1016/j.iot.2019.100056
2542-6605/© 2019 Elsevier B.V. All rights reserved.

The scenario is exacerbated by the unpredictable security consequences resulting from the integration of smart devices into the common Internet services, since the majority of Internet technologies and communication protocols were not designed to support IoT. For these reasons, identifying proper threats for a specific IoT deployment and carrying out a risk assessment process in order to evaluate existing risks and identify related countermeasures are very complex tasks, often very expensive and resource-demanding compared to the cost and time needed to acquire and set-up an IoT system for low cost applications, as smart homes.

In light of the above, in this paper we present a methodology aimed at supporting the security analysis of an IoT system by means of an *almost completely automated process for threat modeling and risk assessment*. The proposed methodology, based on an extension of the approach proposed in [3], relies upon a modeling approach to represent both the architectural components of an IoT system and its security properties, and enables to (i) identify applicable threats, (ii) analyze and evaluate related security risks, and (iii) select the proper countermeasures to enforce in order to mitigate existing risk.

According to the proposed process, the analysts are supported in the definition of a high-level architectural model of the IoT system under analysis in compliance with up-to-date standard specifications, and are only required to provide some technical information on the implementation/deployment of involved components (i.e., devices, services, networks. etc.). Based on the system specification, the threat model is automatically built thanks to the information stored in a security knowledge base (i.e., the *threat catalogue*) that suitably maps threats to assets, countermeasures and other relevant information. The identified threats are then associated with a risk level, computed according to the OWASP Risk Rating Methodology [4], and mapped to a set of suitable countermeasures, in terms of security controls to enforce in the system, which represent the means to mitigate existing risk.

The remainder of this paper is structured as follows: in Section 2, we discuss the characterization of IoT systems based on available international standards and present the state of art related to the analysis of security and threat modeling issues in IoT. In Section 3, we summarize the motivations behind our work and present an overview of the proposed methodology, by also introducing a home automation case study application that will be used throughout the paper to exemplify our proposal. In the subsequent three sections, we describe in detail the three main steps of the methodology, namely *system modeling, threat modeling* and *risk analysis and security control identification*. Finally, in Section 7, we draw our conclusions and discuss some future direction.

## 2. Related work

The exponential growth of IoT devices foreseen for the next future (up to 25.1 billion units in 2021, according to Gartner [5]), along with the evidence of the risks posed by the weak security measures typically adopted in present-day units [6] have made IoT security a very hot topic. Unfortunately, no solid and stable security solutions currently exist. The novelty of the problem, the “volatility” of the devices’ hardware and software architectures and even the consumer behavior [7] call for further research.

The first issue encountered when carrying out an analysis of IoT security issues is the complexity of the current IoT landscape [8]: as anticipated in fact, it is characterized by the coexistence of heterogeneous technologies, which undergo a continuous evolution with the technical advances proposed by existing vendors, and by the application of IoT systems in very different domains.

In order to cope with this issue, several efforts have been recently spent, in both academia and industry, towards an effective characterization of IoT systems. In particular, in the academic field, several survey papers have been recently published (such as [9–11], etc.) that offer a review of the main IoT applications, enabling technologies, architectural requirements, network topologies and device platforms, and which propose different architectural views of IoT systems. On the industrial side, there are many initiatives promoted by the main commercial players and standardization bodies for the definition of a shared IoT reference architecture. Among them, it is worth mentioning the *Industrial Internet Reference Architecture* (IIRA), promoted by the Industrial Internet Consortium (founded by AT&T, Cisco, General Electrics, IBM and Intel) and released in its last version in 2017 [12], and the *Internet of Things Architecture* (IoT-A) proposed by the IoT-A FP7 project [13]. Moreover, some big vendors have released their own architectures in the form of white-papers (e.g., Microsoft, SAP, Intel). Furthermore, it is worth noting that there is an active standardization effort in the direction of defining an IoT Reference Architecture by ISO, i.e., the *ISO Internet of Things Reference Architecture* (IoT RA – ISO/IEC WD 30141), and by IEEE, i.e., the *IEEE Standard for an Architectural Framework for the Internet of Things* (IoT), proposed by the IEEE P2413 WG [14].

As outlined in [1,15], a single reference architecture may not be adequate for all conceivable environments and applications, while the combination of more reference architectures may be adopted to provide a comprehensive and effective IoT vision. In particular, as pointed out in [1], the stack proposed by Borgia [10], which explicitly discriminates among a short-range communication layer (used for the interactions among low-power IoT devices) and a high-bandwidth communication layer (used for the interactions with the upper services) may be used in combination with the standard ISO IoT RA to model the most critical features of all the proposed reference architectures.

It is worth noting that the availability of a single, homogeneous and comprehensive description of IoT systems is a fundamental requirement to drive the security assessment process of an actual IoT deployment. Even if, as pointed out in [15], some of the above IoT architecture specifications do include security considerations, security is typically considered as a vertical layer, something that can be managed separately and that cuts across multiple architectural layers. As demonstrated

by the number of security incidents that are reported daily, a more reliable model should be adopted, devising specific solutions (techniques and mechanisms) tailored to the peculiar features of each layer.

The main security issues affecting the IoT world have been widely discussed in several recent papers, including [15–19], which provide a good picture of the general IoT threat model. The authors of [20] consider a three-layer view of an IoT system (i.e., application, perception and network layer) and provide a wide overview of existing security threats and vulnerabilities in the IoT environment as opposed to traditional (wireless) systems. They propose an IoT security taxonomy that involves in general the applications, architectures, communications and data, and suggest some possible solutions for improving the IoT security. Similarly, the authors of [21] propose a systematic view of IoT systems, by identifying the main involved elements and their interconnections, the main actors and their relationships in the IoT context, and analyze the existing security challenges with respect to each element and actor. Other interesting reviews are provided by the slightly older papers [22] and [23], which study IoT security challenges in multiple security domains (e.g., authentication, access control, privacy, fault tolerance, etc.), proposing an overview of security threats and open issues. It is worth mentioning that, in the current literature on IoT security, a relevant role is played by papers that address specific threats against well-defined protocols/platforms. For example, Kasinathan et al. present in [24] a denial of service (DoS) detection architecture for 6LoWPAN, a protocol designed by IETF and enabling low-power devices to communicate with the Internet, while the authors of [25] present an intrusion detection approach relying upon an Artificial Neural Network (ANN) to identify DoS attacks in a generic IoT system.

Despite the high number of research efforts spent toward the security characterization of IoT systems, there is no comprehensive approach that enables to properly capture the actual security issues existing in a real IoT deployment, to be taken into account for security assessment purposes. An interesting approach, relying upon a technique with some points in common with the one presented in this paper, is represented by the recent work by Lewis [26], which evaluates a risk profile of a system configuration by leveraging graphs and graph databases to model and process information related to IoT devices' functional and security properties. The main difference between the above approach and ours is that the authors of [26] do not use a standard reference architecture to model IoT systems and adopt simple empirical risk metrics and threshold values to obtain the IoT system risk profile, while our approach founds on a standard IoT characterization, relies upon widely accepted methodologies for risk assessment, and adopts a complex security model represented by a threat catalogue built by gathering information about several well-known threats and vulnerabilities affecting IoT systems at several layers.

In summary, the main limitation of existing proposals in the field of security analysis and risk assessment of IoT systems is the lack of a reference standard for describing the IoT systems and modeling their security properties. Most of the existing proposals describe an IoT system based on a single, often simplistic and only partially correct reference architecture, and typically refer to specific security domains and security taxonomies. As a consequence, the resulting threat models can be hardly reused and/or integrated in order to build up a single coherent IoT threat model. As discussed in the remainder of this paper, our proposal aims at bridging this gap by providing a standard-oriented process to support the security assessment of an IoT deployment, which requires only a very limited human intervention and that enables to dramatically reduce the costs of security design. As anticipated, this paper is an extension to our preliminary work presented in [3], where we introduced the general methodology and the main techniques and tools behind the process. Compared to our previous work, this paper provides a broader discussion of IoT systems and security models with reference to existing standards, by illustrating more in details each phase of the proposed process and by discussing it with a complete case study.

## 3. Motivation and proposal

As discussed in Section 2, the lack of a commonly accepted IoT model makes it more complex to identify the assets to protect in an IoT deployment, since each modeling solution may stress a different aspect while not providing a complete view of the system. As anticipated in the previous section and discussed in detail in [1], the ISO standard can be fairly used to map the concepts of most part of the other relevant IoT standards, and its effectiveness can be further enhanced by explicitly taking into account one of the peculiarities of IoT systems, namely the typical coexistence of different communication networks (i.e., a low-power proximity network among edge devices and a high-bandwidth network to connect devices with Internet services), characterized by different technologies, performance and security features.

A further problem is represented by the lack of a comprehensive threat model for IoT systems, able to take into account the peculiar features of all the possible components of a complex IoT infrastructure. This shortage makes it very difficult to perform an effective security assessment of actual IoT deployments. In fact, while the literature on threats for specific solutions and technologies is quite generous, it is very hard to have a complete and coherent list of applicable threats which are specific to a system to be deployed in production.

Last but not least, it is fundamental to take into account who are the main actors involved in the set-up and deployment of an IoT system. It is worth noting, in fact, that such activities are usually delegated to technicians without specific expertise, since the involvement of highly-skilled (and expensive) security experts is often not economically viable. To make an example, consider the case of the deployment of a smart home system: it is connected to the home network offered by the provider and gives full control over the house. Although this opens to a lot of risks, no one would involve a skilled security expert to configure such a system upon installation.

To address these issues, we propose a methodology that, relying upon a modeling approach that takes into account the ISO standard directives, enables to build, in a semi-automated way, the threat model for a specific IoT system deployment,

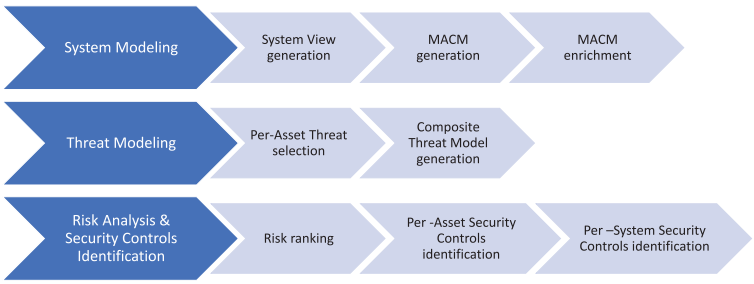


**Fig. 1.** IoT automated risk analysis methodology.

and to support the secure design activity by determining the set of security countermeasures to enforce to mitigate existing risk. Security countermeasures, in particular, are specified in terms of security controls, defined according to the NIST Security Control Framework [27]. The mapping onto other relevant existing frameworks (such as the ISO 27001 framework) is quite straightforward and already available, which makes our approach flexible and easily reusable in different contexts.

The adoption of standards enables to apply our methodology to multiple heterogeneous technologies, and the automation of the risk assessment process enables to achieve a minimal security level even when the analysis process is performed by a technician not expert in the security field.

### *3.1. Proposal overview*

Building on top of the standard IoT characterization discussed above, the methodology presented in this paper enables to perform threat modeling and risk assessment of IoT systems in an (almost completely) automated way. The proposed methodology, sketched in Fig. 1, comprises three main steps:

- *System modeling*: it is devoted to analyzing the IoT system in order to identify and suitably model the main assets to protect.
- *Threat modeling*: it is devoted to identifying all relevant threats for the system.
- *Risk analysis and security controls identification*: it is devoted to estimating the risk associated with each identified threat, and to determining the countermeasures that must be put in place in order to mitigate existing risks, expressed in terms of standard security controls.

In the following sections, we will detail each of the above phases.

### *3.2. The MicroBees case study*

In order to illustrate the proposed approach, we will consider a home automation system built by leveraging the MicroBees IoT technology [28]. MicroBees offers a set of components (a gateway, *gateBee*, and two sensor/actuator devices, *senseBee* and *wireBee*) devoted to offering simple and cheap home automation functionalities. As sketched in Fig. 2, such components interact via radio by means of a custom protocol, and are coordinated by a dedicated gateway that adopts cloud services to offer advanced user interface and improved automation capabilities. The end user interacts with the system through a mobile phone, by accessing the cloud services that communicate with the gateway component.

It is worth noting that this case study catches all the issues we outlined before: the solution aims at being cheap and easy to install, so the technician devoted to installing is non highly skilled and typically completely unaware of security issues. Moreover, it is configured inside an home network and works on top of it. In the remainder of the paper, we will use this simple case study to illustrate how it is possible to adopt the proposed methodology to carry out a fast security assessment on such a solution, granting the right level of awareness to the owner, who may apply, in such a way, the right countermeasures (possibly by involving an expert).

## 4. System modeling

The system modeling phase is devoted to building a model of the IoT system to be deployed in compliance with the ISO standard, whose main features are summarized in Section 4.1. The model is built by using a graph-based formalism, discussed in Section 4.2, which provides a simple yet powerful means to represent the information on the system architecture and on the involved assets' properties that enables the automation of most part of the security assessment process.

### *4.1. IoT systems characterization according to ISO*

The ISO IoT Reference Architecture (IoT RA), described in the ISO/IEC 30141 specification, includes five different views. The *functional view* provides a technology-agnostic view of the functions necessary to form an IoT system and of their interdependencies. The *system view* describes the architecture of an IoT system in terms of its physical components (e.g., devices,

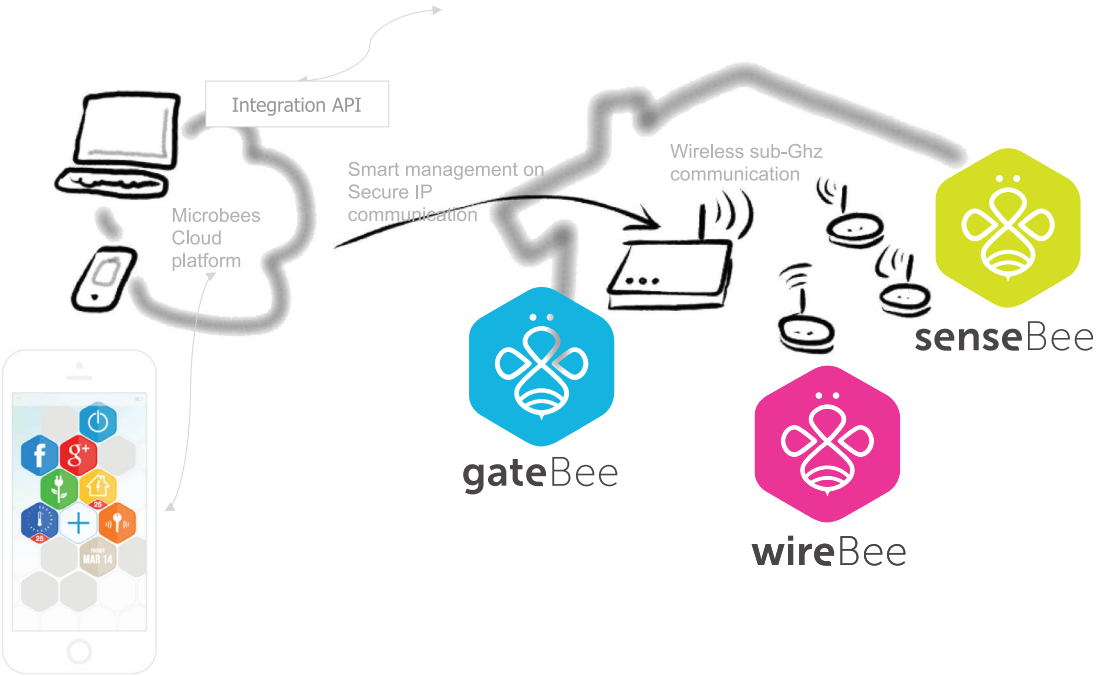


**Fig. 2.** The MicroBees IoT solution.

sub-systems, networks), their distribution, their interconnections, and the related behavior and properties. The *communication view* describes the main communications networks involved in an IoT system (these are classified in user network, service network, access network and proximity network). The *information view* involves the data generated by using, monitoring, controlling and analyzing connected entities. Finally, the *usage view* provides the details on how the IoT system is developed, tested, operated and used from a user perspective.

All the above views are organized based on six *domains*, which have been identified based on a study of the decomposition of IoT systems in different application scenarios. These include: the User Domain, which is represented by the end-users of the system, the Operations & Management Domain, the Application Service Domain, which collects the services offered to end-users, the Resource & Interchange Domain, the Sensing & Controlling Domain, which is composed of the sensors, actuators, and gateways and the Physical Entity Domain, which includes the physical and virtual entities of a system.

The ISO IoT RA specifies the main characteristics of a generic IoT system through a conceptual model (CM) that describes the entities involved in an IoT system along with their relationships. Among the main entities defined by the CM, sketched in Fig. 3, it is worth mentioning the *IoT Device*, the *IoT User*, the *Application*, and the *Network*. In particular, an *IoT Device* is the entity that bridges between real-world *Physical Entities* and the other digital entities in the system. An IoT Device can be either a *Sensor*, able to monitor a physical entity and transform some of its characteristics into a digital representation that can be communicated, or an *Actuator*, able to act on one or more properties of a physical entity on the basis of received commands.

The *IoT user* is the human or digital entity that uses the IoT system by means of an *Application* or of a set of *Services*, respectively. A Service is able to manage *Virtual Entities*, which are the digital representations of the physical entities of the system, and may use a *Data Store* holding the data generated by IoT Devices or other Services.

IoT Devices typically interact with Services by means of an *IoT Gateway*, and the connections are enabled by a *Network*. It is worth noting that a Gateway usually interacts with IoT Devices through short-range networks, while the communication with Services typically involves high-bandwidth networks; from a security perspective, this consideration is very critical and should be taken in consideration while building the IoT threat model.

The concepts defined by the standard can be used to model every type of technology involved in a generic IoT system deployment, thus coping with the heterogeneity issue that usually represents an obstacle to the security assessment process. To remark this concept, let us consider again our case study, represented by the MicroBees home automation application. The MicroBees technological solution includes the following components:

- *WireBee:* it is responsible for monitoring and measuring different physical features;
- *SenseBee:* enables the automation of different sensing rules;
- *GateBee:* it is the central gateway that receives commands and data; it communicates with SenseBee and WireBee via a radio channel;

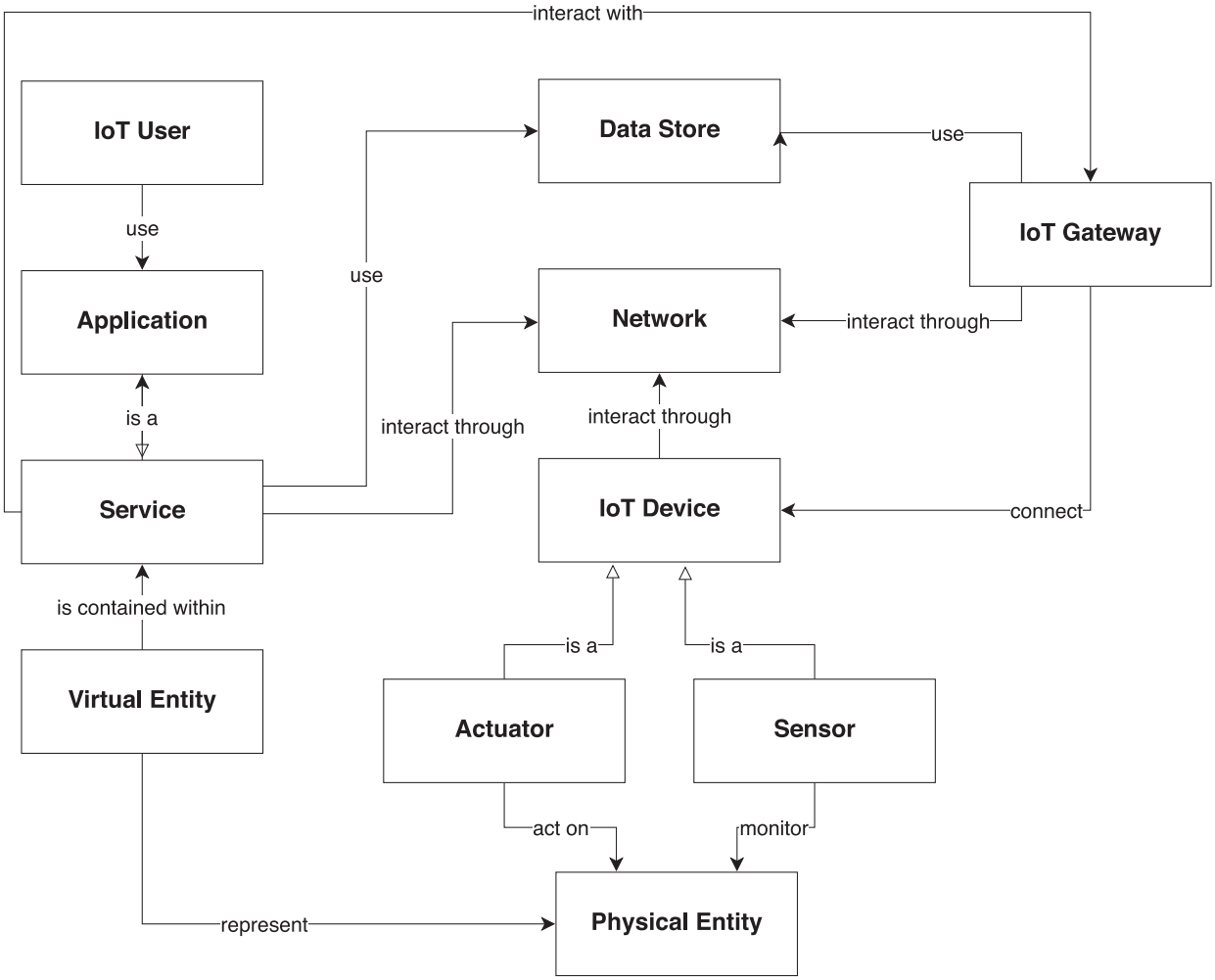


**Fig. 3.** ISO IoT concepts.

Taking into consideration the ISO conceptual model, we can simply map the specific MicroBees technology onto ISO concepts as illustrated in Fig. 4. The light yellow classes represent the specific MicroBees components.

Since the security properties of an IoT system heavily depend on its deployment and on related technological aspects, we adopt the ISO *system view*, which specifies the system components and their interconnections, as the starting point for security analysis. Hence, the first activity of the *System Modeling* step consists in building the system view of the IoT system under analysis (*System View generation* sub-step) by identifying, in particular, the components belonging to the Sensing & Controlling domain (i.e., sensors, actuators, gateways), the components of the Application Service domain (i.e., the services that enable data access, processing, and storage, identity resolution, geographic information service, user management, etc.), and the specific networks involved in components communications along with network appliances. It is worth mentioning that, with regard to the Application Service domain, in this paper we consider services running on cloud infrastructures, since this is a common scenario in current IoT systems.

In order to model a real use case, let us consider a MicroBees-based home automation system devoted to managing the lighting and heating of a private house through a set of actuators (represented by the `SenseBee` devices) controlling garden lights, entrance lights, kitchen lights and a thermostat, respectively, and one sensor (represented by the `WireBee` device) that acts as a thermometer. The `SenseBee` and `WireBee` devices are connected via radio to the `GateBee` device, which is the central gateway and enables the access to the MicroBees cloud service. The user interacts with the system through a custom application accessed via his/her mobile phone.

As said, the first step of *System Modeling* consist in building the ISO-compliant system view of the IoT system under analysis, which specifies the main involved assets (physical components, services and networks). The system view built for the MicroBees home automation system is depicted in Fig. 5, which shows the components belonging to the ISO User

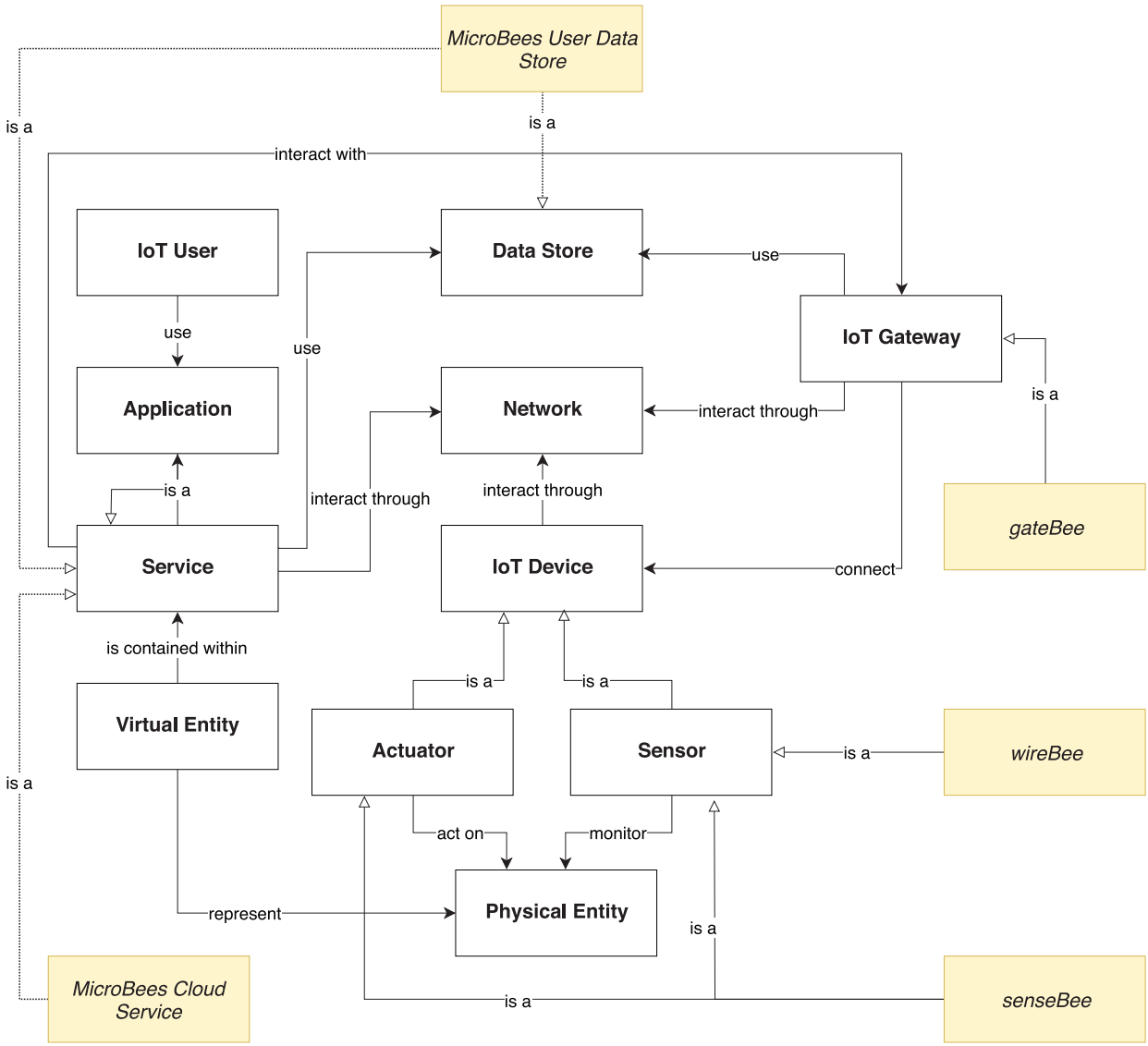


**Fig. 4.** MicroBees reference architecture in the ISO model.

Domain (UD), Application Service Domain (ASD), Sensing & Controlling Domain (SCD) and Physical Entity Domain (PED), and those used for component communications (networks and network appliances).

### *4.2. The modeling process based on the IoT MACM formalism*

In the *MACM generation sub-step*, the system view of the IoT system under analysis is automatically translated into a graph-based formalism, named *IoT Multi-cloud Application Composition Model* (IoT MACM). The MACM formalism, introduced in [29], was initially proposed in the context of multi-cloud applications and was aimed at representing their architectural aspects and their security properties, defined in terms of standard security controls included in a Security Service Level Agreement (Security SLA). As discussed in [29,30], by means of suitable manipulations of the MACM representation of an application, it is possible to perform an automated security assessment that enables to identify the Security SLA that can be actually granted (i.e., the security controls that can be considered as correctly implemented) by each application component, based on the knowledge of how components are implemented internally and on their interactions.

The IoT MACM extends the MACM formalism and defines a set of *node types* and *relationships* that enable to model a generic IoT system in compliance with the ISO IoT RA concepts. The considered node types are reported in Table 1. The relationships introduced by the IoT MACM formalism are summarized in Table 2 that reports, for each relationship, the admissible start and end node types.

**Table 1**
IoT MACM node types.

| Node Type | Description |
|---|---|
| *CSP* | Service Providers that offers generic cloud services |
| *Service* | Service belonging to the ISO Application Service domain |
| *Device* | Physical device representing either (i) a component belonging to the ISO Sensing & Controlling domain (i.e., an IoT device acting as sensors/actuators or another type of device used to collect data), (ii) a user interface device belonging to the User domain, or (iii) a network equipment appliance used to enable communication |
| *IoTGateway* | Physical device that acts as a gateway between a subset of Device nodes and the upper Services, by typically providing advanced data collection, aggregation and filtering functionalities |
| *Network* | Physical network connecting system components |

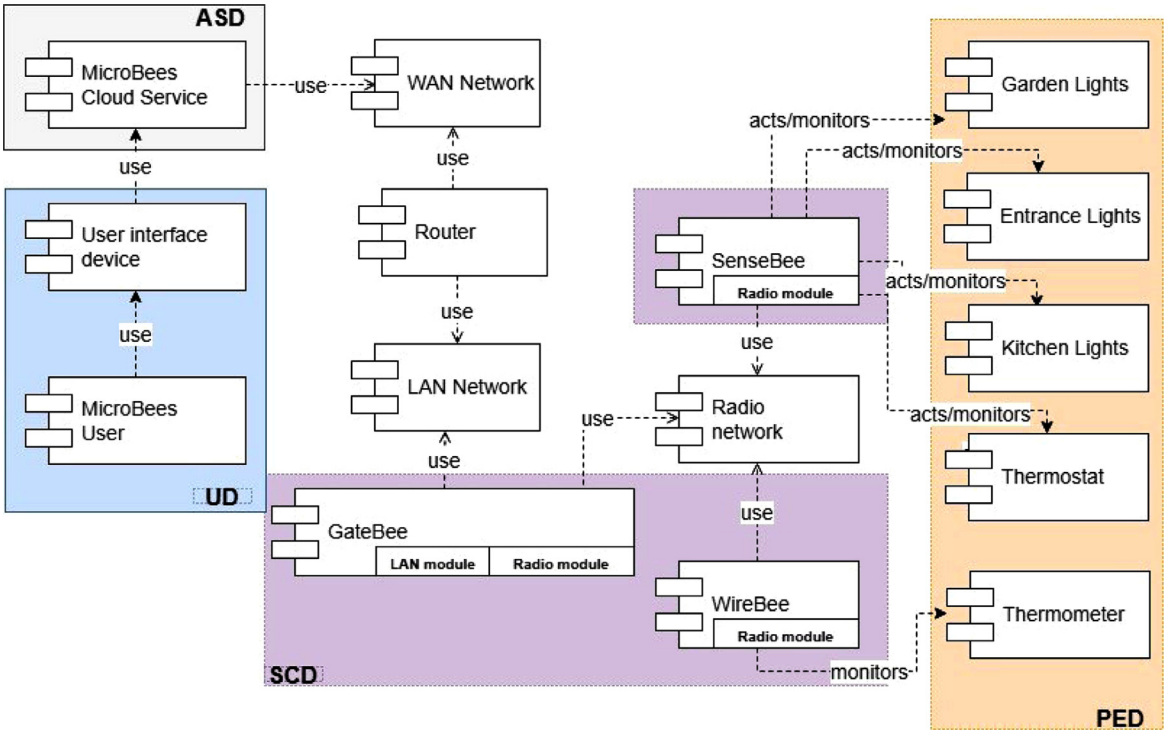


**Fig. 5.** MicroBees home automation system – system view.

In the *MACM enrichment sub-step*, the MACM model of the target system is enriched with additional information useful to better characterize the involved assets and to identify the threats potentially affecting them. This information can be obtained by resorting to existing industry or research-oriented reconnaissance tools or, when these are not enough, by querying a human assessor. The MACM model of the target system, in particular, is enhanced by specifying for example the type of an asset, where needed (e.g., is a network asset a radio network, LAN or a WAN?, is it a wired or a wireless network? is a service asset a web-based service?, is an IoT device an open-platform device?, etc.), the type of protocol used in a communication (e.g., XMPP, Zigbee, TLS/SSL, IP, HTTP, HTTPS, TCP), the role of a node in a communication protocol (e.g. server node, client node, peer node, etc.), and the behaviour of a node (e.g., does a web service allow users to input data?). These details are used to annotate the model by means of suitable node and edge properties.

Fig. 6 shows the MACM model for the MicroBees case study. As said, the MACM model includes different node types for the different concepts of the ISO model. Some of the nodes model specific MicroBees components (gateBee, senseBee, wireBee), but also other components, which are not part of the MicroBees technological stack, are modeled (e.g., the router and/or the user mobile device). The MACM representation, in addition, models the relationships and the interconnections among components. The three networks adopted in the system deployment put in evidence how components communicate and, as we will see later, will help identify the possible threats in the architecture.

**Table 2**
IoT MACM node relationships

| Relationship | Start Node Type | End Node Type | Description |
|---|---|---|---|
| *provides* | CSP | Service | Describes a cloud service provider that offers infrastructure services (i.e., virtual machines) |
| *uses* | Service | Service, Device, IoTGateway | Describes a generic Service that may use another Service (i.e., invoke its functionalities) or directly access the functionalities offered by an IotGateway or a Device (e.g., a Map Service may query a set of traffic sensors to retrieve specific traffic information related to a given time frame) |
| | IoTGateway | Device, Service | Describes an IoTGateway that uses a Device node (e.g., by sending queries and commands) or an external Service |
| | Device | IoTGateway, Service | Describes a Device that uses an IoTGateway (e.g., to forward collected data or communicate an alert), and that may also use (i.e., directly invoke) a Service (e.g., a smart thermostat may access a weather forecast service to change dynamically its settings based on hourly forecasts) |
| *hosts* | IotGateway | Service | Describes an IotGateway that hosts a dedicated Service (e.g., a user-defined application running on a Raspberry gateway node) |
| | Device | Service | Describes a generic Device that hosts a dedicated Service (e.g., the Cisco IOx applications running on smart networking appliances) |
| *connects* | Service | Network | Specifies the Network infrastructure through which the Service is invoked |
| | IoTGateway | Network | Specifies the network infrastructure to which an IoTGateway node is physically connected |
| | Device | Network | Specifies the network infrastructure to which a generic Device node is physically connected |

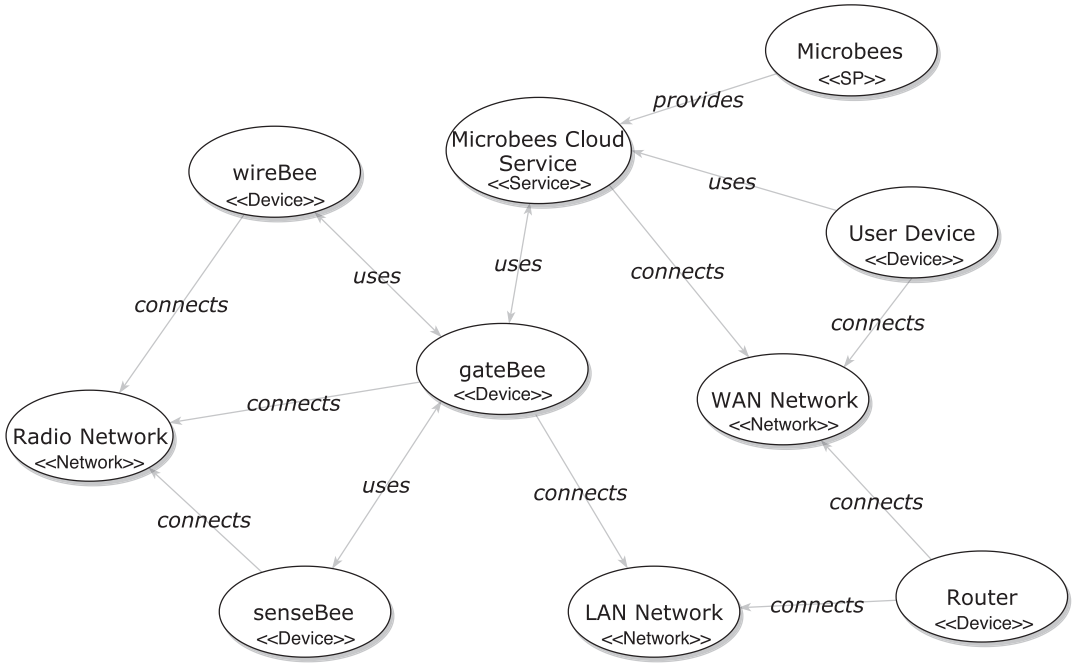


**Fig. 6.** MicroBees home automation system – MACM model.

## 5. Threat modeling

The *Threat Modeling* step is devoted to building a threat model for the system under analysis. While the *System Modeling* step does need a human intervention for building the system model and replying to the questions on involved assets, this step can be completely automated thanks to the availability of a knowledge base.

Such knowledge base is represented by a *threat catalogue*,[1] which has been developed and enriched in the context of the FP7 SPECS project (www.specs-project.eu) and of the MUSA H2020 project (www.musa-project.eu), and that collects several

[1] www.bitbucket.org/cerict/sla-model

**Table 3**
Extract of the threat catalogue.

| Threat | Description | Asset | Asset type | STRIDE | CIA | Security controls |
|---|---|---|---|---|---|---|
| Eavesdropping | An adversary can retrieve sensitive data by observing the communication channel | Network | Radio | Information disclosure | Confid. | AC-4, AC-16, AC-17, SC-7, SC-8, SC-10, SC-12, SC-13, SC-17, IA-2, IA-7 |
| Data Leakage | An adversary can access local data of the asset | Device, IoTGateway, SaaS | Peer, Client, Server, CMS | Information disclosure, spoofing | Confid. | AC-7, AC-19, IA-3, IA-3(1), IA-5, SA-18, SC-8, SC-41, SI-2, RA-5(1) |
| Message Modification | An adversary can intercept and modify communi-cationpackets | Network | Radio | Information disclosure, spoofing, tampering | Confid., Integrity | AC-16, AC-17, SC-8, SC-13, SC-17, SC-23, SC-38, SC-40, SA-18 |
| ... | ... | ... | ... | ... | ... | ... |

**Table 4**
Extract of threat model for the MicroBees deployment.

| System component | Asset type | Threat |
|---|---|---|
| GateBee | IoTGateway | Data leakage |
| GateBee | IoTGateway | Topology disclosure |
| Radio Network | Network | Message modification |
| Radio Network | Network | Message injection |
| SenseBee | Device | Data leakage |
| SenseBee | Device | Exaustion of power |
| User Device | Device | Data leakage |
| Microbees Cloud Service | Service | Denial of service |
| ... | ... | ... |

well-known threats affecting different assets, including software components and communication protocols (we currently support Ethernet, IP, TCP, TLS, XMPP, OAUTH, Zigbee, and Bluetooth, and we are continuing updating the threat collection). In the catalogue, threats are classified based on the asset they apply to, taking into account the specific type of asset and its behavior. In particular, the main assets identified by ISO were considered, and assets were classified based on the node types specified by the IoT MACM formalism. Moreover, threats are associated with the STRIDE category [31] they belong to (i.e., Spoofing, Tampering, Repudiation, Information-Disclosure, Elevation-of-Privileges) and with the CIA (Confidentiality-Integrity-Availability) property they undermine. Finally, threats are mapped to the standard security controls that represent the countermeasures to put in place in order to mitigate the risk deriving from their realization. We considered in particular the NIST Security Control Framework [27], which lists more than 900 security controls applicable to several aspects of an IT system, but other frameworks may be considered as well. In Table 3 we report an extract of the catalogue.

During the *Per-Asset Threat Selection* sub-step, the threats applicable to each component of the system (namely, to each involved asset) are identified. In particular, based on the asset type and on related additional information collected during the modeling step (including the relationships among the components and the adopted communication protocols), applicable threats are retrieved automatically from the catalogue.

The set of threats identified for all system assets represents the IoT system's threat model, which is the output of the *Composite Threat Model generation* sub-step.

We applied the above process to the MicroBees case study and obtained a threat model including about one hundred different applicable threats. Table 4 shows a very small extract of the achieved results due to an existing non-disclosure agreement with the MicroBees company. Each row of the table reports the system component, the respective asset type and one associated applicable threat.

## 6. Risk analysis and scurity controls identification

The *Risk Analysis and Security Control Identification* step is devoted to computing an estimation of the risk associated with each threat identified at the previous stage, and to determining the security controls to implement to protect each asset in order to mitigate such risk.

The first activity, conducted in the *Risk Analysis* sub-step, follows the OWASP Risk Rating Methodology [4], which adopts 16 different parameters related to the *likelihood* of having the threat realized and to the technical and business *impact* of this realization on the system. In particular, likelihood parameters are classified into *threat agent factors* (including Skill level, Motive, Opportunity and Size) and *vulnerability factors* (including Ease of discovery, Ease of exploit, Awareness and Intrusion detection), while impact parameters take into account both *technical factors* (related to Loss of confidentiality, Loss

**Table 5**
The OWASP risk rating methodology: determining risk severity.

| Overall risk severity | | | | |
|---|---|---|---|---|
| *Impact* | HIGH | Medium | High | *Critical* |
| | MEDIUM | Low | Medium | High |
| | LOW | Note | Low | Medium |
| | | LOW | MEDIUM | HIGH |
| | *Likelihood* | | | |

of integrity, Loss of availability and Loss of accountability) and *business factors* (related to Financial damage, Reputation damage, Non-compliance and Privacy violation).

According to the methodology, the security analyst is asked to assign a numeric value to each of these parameters in a range from 0 to 9. This task is made simpler by the definition of a set of pre-defined options for each parameter, each having a pre-defined numeric value assigned. For instance, let us consider the 'Skill level' threat agent factor considered by the methodology: for this factor, which represents the level of skill that a set of threat agents should have to realize a given threat, the methodology suggests five different options: 'no technical skills' (value 1), 'some technical skills' (value 3), 'advanced computer user' (value 5), 'network and programming skills' (value 6), and 'security penetration skills' (value 9).

In our process, in order to limit human interaction and facilitate the analysis, 12 of the parameters adopted by the OWASP methodology (i.e., likelihood parameters and technical impact parameters) are pre-evaluated for each threat in the catalogue and assigned default values that are automatically associated with the threats identified in the *Threat modeling* step. The only parameters the Analyst has to manually evaluate are the ones related to business impact.

Once a numeric value has been assigned to each factor, the overall likelihood and impact level are computed as the average of the values of respective factors. Afterward, the obtained numeric value is converted to a qualitative level in [LOW, MEDIUM, HIGH]. In particular, values in [0;3[ correspond to a LOW level, values in [3;6[ correspond to a MEDIUM level, and values in [6;9] correspond to a HIGH level.

Finally, the final risk associated with a threat is obtained by suitably combining respective likelihood and impact levels based on a predefined match table (like the one shown in Table 5). Of course, the match table should be tuned based on the business context to make better risk decisions. Let us consider for example the case of an evaluation resulting in a medium likelihood and in a high impact, with the latter being composed by a negligible business impact and a severe technical impact. According to Table 5, the overall risk severity is to be considered HIGH but, in specific contexts, it may be more properly evaluated to LOW due to the poor impact on business.

It is worth noting that, in order to simplify the process, the analysis may be conducted on groups of threats classified based on the respective STRIDE category rather than on each threat separately.

After the *Risk Analysis* sub-step, each threat that has been found applicable to system assets is assigned a risk value in {`LOW,MEDIUM,HIGH`}. Based on this values, the subsequent *Per-Asset Security Controls identification* sub-step takes place, which is aimed at identifying the appropriate set of security controls that should be implemented at each asset based on the level of risk assigned to applicable threats. In fact, we adopt an approach similar to the one suggested by the NIST Control Framework, which proposes a specific risk analysis process and suggests the adoption of security control *baselines* for low-impact, moderate-impact, and high-impact information systems, respectively. In practice, for each risk level, the associated baseline identifies the security controls that should be implemented in a system to which *at least* that risk level has been assigned. Note that the three baselines are hierarchical: if a control belongs to the LOW impact baseline, then it will also appear in MEDIUM and HIGH impact baselines. As an example, NIST security control AC-12 belongs to the MEDIUM impact baseline (while it is not selected in the LOW-impact baseline): this means that this controls must be implemented, if required, not only in the systems with MEDIUM risk severity, but also in systems with HIGH risk severity.

Compared to the NIST framework, we rely on a different risk analysis process, since we assign a level of risk to a single threat rather than to a system, but we use the same control baselines that enable to select appropriate security controls in a completely automated way.

Let $c$ be one of the assets identified for the considered IoT system, $T_c = \{t_1, \ldots, t_N\}$ be the set of threats found applicable for asset $c$ as a result of the *Threat Modeling* step, and $SC(T_c) = \{sc_1, \ldots, sc_M\}$ be the set of all security controls that represent possible countermeasures for the threats in $T_c$, obtained by simply querying the catalogue.

The *Per-Asset Security Controls identification* sub-step consists in properly selecting the security controls $SC_c \subset SC(T_c) = \{sc_1^c, \ldots, sc_K^c\}$ to implement at asset $c$ based on the risk severity associated with applicable threats $T_c$.

Let us denote with $risk(t_i)$ the level of risk assigned to threat $t_i \in T_c$ by the *Risk Analysis* sub-step, and let us denote with $min_risk(sc_j)$ the *minimum* system risk level (i.e., the lower risk baseline) for which the NIST recommends the implementation of control $sc_j$.

The following condition holds:

$$sc_j \in SC_c \Leftrightarrow \exists t_i \in T_c : risk(t_i) \geq min_risk(sc_j) \quad \forall sc_j \in SC(T_c) \tag{1}$$

**Table 6**
Notation summary.

| Symbol | Description |
|---|---|
| $c$ | Asset of the system |
| $T_c = \{t_1, \ldots, t_N\}$ | Set of threats found applicable for asset $c$ |
| $SC(T_c) = \{sc_1, \ldots, sc_M\}$ | Set of all security controls that represent possible countermeasures for the threats in $T_c$ |
| $SC_c = \{sc_1^c, \ldots, sc_M^c\}$ | Set of the security controls to actually implement at asset $c$ based on the risk analysis |
| $risk(t_i)$ | Level of risk assigned to threat $t_i \in T_c$ by the risk analysis |
| $min_risk(sc_j)$ | Minimum system risk level for which implementation of control $sc_j$ si recommended |

**Table 7**
Extract of the list of security controls to implement in the MicroBees deployment.

| Asset | Security controls |
|---|---|
| GateBee | IA-3, IA-3(1), SA-18, SC-41, IA-5, SC-8, SI-2, RA-5, AC-2, AC-1, AC-7, AC-9, IA-5(1) |
| Radio Network | AC-17, SC-8, IA-2(13), SC-23 |
| SenseBee | IA-3, IA-3(1), SA-18, SC-41, IA-5, SC-8, SI-2, RA-5(1), AC-2, AC-1,SC-41, AC-7, AC-9, IA-5, SC-8, IA-5(1) |
| User Device | AC-7(2), AC-19, IA-3, IA-3(1), SA-18, SC-41, IA-5(1), IA-5, SC-8, SI-2, RA-5(1) |
| Microbees Cloud Service | IA-9, SA-18, AC-2, AC-1, AC-7, AC-9, IA-5, SC-8, IA-5(1), SI-2, RA-5(1) |
| ... | ... |

The above equation should be read as follows: a control $sc_j$ belonging to the lower risk baseline $min_risk(sc_j)$ must be implemented at asset $c$ if and only if there is at least one threat $t_i$ affecting asset $c$ to which a risk equal or greater to $min_risk(sc_j)$ has been assigned.

The symbols introduced above are summarized in Table 6 for clarity sake.

To better clarify the concepts introduced above, let us consider a control $sc$ representing a countermeasure for two threats $t_1$ and $t_2$, both associated with asset $c$, and let us assume that $t_1$ has been assigned a LOW risk severity while $t_2$ has a MEDIUM risk severity. If $sc$ appears in the MEDIUM baseline, it means that it must be enforced in any system with at least MEDIUM risk severity. Since $t_2$ has such as severity level, control $sc$ will be selected as a result of this step of the process. If instead $sc$ appears only in the HIGH baseline, there is no need to include it as none of applicable threats has been assigned a HIGH risk.

At this point, we have a mapping between each asset of the system and the controls that should be implemented locally, based solely on the threats affecting the asset and on respective risk level. It is worth noting that this mapping may be not complete, since the threat modeling is carried out independently for each asset of the system without considering the mutual relationships among components and the impact of their deployment on cloud resources. Let us assume, for example, that the system under analysis includes a service S that uses and configures a gateway G, which in turn uses a sensor device D to get environment-related data. Let us assume that, by following the methodology described above, it results that control AC-3 (requiring the enforcement of an access control mechanism) must be implemented at component G. It is important to note that, in order to consider the control as effectively implemented at component G, it must be also enforced on the service that makes access to G to configure it, and thus the control must be added to those identified for S. Taking these considerations into account, a further step is needed aimed at evaluating the need for additional security controls at each asset of the system based on those identified in the *Per-Asset Security Controls identification* sub-step and on the relationships existing among the components. This step is named *Per-System Security Controls identification*, is carried out by performing specific reasoning over the MACM representation, as widely discussed in [29,30], and allows to obtain the actual list of controls to implement in each asset of the system based on the relationships existing among assets and on the behavior of used (cloud) resources. We invite the interested reader to take a look into the existing literature to have the details of the mentioned process.

Table 7 reports some of the security controls recommended for application for a subset of the components of the Microbees case study, resulting from the application of the discussed threat modeling and risk assessment process. Note that the list of controls provided in output to the process is a concrete checklist that a technician, even with limited security competences, can verify to assess that the existing risks are correctly mitigated.

## 7. Conclusions and future works

In this paper, we introduced a methodology to support the threat modeling and risk analysis of IoT systems by means of an almost completely automated process. Starting from the system model, built in compliance with the ISO/IEC 30141 standard directives, and thanks to the information collected in a threat catalogue, the proposed process enables to (i) identify applicable threats, (ii) evaluate the risk associated with such threats, and (iii) determine the countermeasures to enforce in terms of security controls.

As a future direction, we plan to extend the process by integrating, on the one hand, automated security assessment features, in order to enable the analysis of the level of security provided by different deployments (e.g., deployments involving

different devices or device configurations, different services etc.) at design time. On the other hand, we plan to integrate automated penetration testing features in the process, based on the framework and solutions proposed in [32].

## Conflict of Interest

No conflict of interest.

## References


[1] B. Di Martino, M. Rak, M. Ficco, A. Esposito, S. Maisto, S. Nacchia, Internet of things reference architectures, security and interoperability: a survey, Interne. Things 1–2 (2018) 99–112, doi:10.1016/j.iot.2018.08.008.
[2] Internet of Things Reference Architecture (IoT RA). ISO/IEC CD 30141:20160910(E). ISO; 2016. Geneva, Switzerland.
[3] M. Rak, V. Casola, A. De Benedictis, U. Villano, Automated risk analysis for iot systems, in: Proceedings of the IPC 2018 Workshop, 2019.
[4] OWASP, The OWASP Risk Rating Methodology Wiki Page, https://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology (accessed 20 May 2019).
[5] I. Gartner, Forecast: Internet of things endpoints and associated services, worldwide, 2017, (https://www.gartner.com/doc/3840665/forecast-internet-things--endpoints).
[6] C. Kolias, G. Kambourakis, A. Stavrou, J. Voas, Ddos in the IoT: Mirai and other botnets, Computer 50 (7) (2017) 80–84.
[7] W.Z. Khan, M.Y. Aalsalem, M.K. Khan, Five acts of consumer behavior: A potential security and privacy threat to internet of things, in: Proceedings of the 2018 IEEE International Conference on Consumer Electronics (ICCE), 2018, pp. 1–3.
[8] M. Weyrich, C. Ebert, Reference architectures for the internet of things, IEEE Softw. 33 (1) (2016) 112–116, doi:10.1109/MS.2016.20.
[9] I. Yaqoob, E. Ahmed, I.A.T. Hashem, A.I.A. Ahmed, A. Gani, M. Imran, M. Guizani, Internet of things architecture: Recent advances, taxonomy, requirements, and open challenges, IEEE Wirel. Commun. 24 (3) (2017) 10–16, doi:10.1109/MWC.2017.1600421.
[10] E. Borgia, The internet of things vision: Key features, applications and open issues, Comput. Commun. 54 (2014) 1–31, doi:10.1016/j.comcom.2014.09.008.
[11] L.D. Xu, W. He, S. Li, Internet of things in industries: a survey, IEEE Trans. Indust. Inf. 10 (4) (2014) 2233–2243, doi:10.1109/TII.2014.2300753.
[12] S.-W. Lin (Thingswise), M. Crawford (SAP) and M. Stephen (IIC), eds. The Industrial Internet of Things Volume G1: Reference Architecture. Report no. IIC:PUB:G1:V1.80:20170131. Industrial Internet Consortium, 2017.
[13] F. Carrez, eds. Deliverable D1.5 Final architectural reference model for the IoT v3.0. IoT-A Consortium, 2013.
[14] IEEE Draft Standard for an Architectural Framework for the Internet of Things (IoT), IEEE P2413, IEEE, 2015.
[15] D. Minoli, K. Sohraby, J. Kouns, IoT security (IoTSec) considerations, requirements, and architectures, in: Proceedings of the 2017 Forteenth IEEE Annual Consumer Communications & Networking Conference (CCNC), IEEE, 2017, pp. 1006–1007, doi:10.1109/CCNC.2017.7983271.
[16] K. Zhao, L. Ge, A survey on the internet of things security, in: Proceedings of the 2013 Ninth International Conference on Computational Intelligence and Security, 2013, pp. 663–667.
[17] Y. Yang, L. Wu, G. Yin, L. Li, H. Zhao, A survey on security and privacy issues in internet-of-things, IEEE Internet Things J. 4 (5) (2017) 1250–1258.
[18] A. Mosenia, N.K. Jha, A comprehensive study of security of internet-of-things, IEEE Trans. Emerg. Top. Comput. 5 (4) (2017) 586–602.
[19] S. Rizvi, A. Kurtz, J. Pfeffer, M. Rizvi, Securing the internet of things (iot): a security taxonomy for iot, in: Proceedings of the 2018 Seventeenth IEEE International Conference On Trust, Security And Privacy In Computing And Communications/twelfth IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), 2018, pp. 163–168.
[20] F.A. Alaba, M. Othman, I.A.T. Hashem, F. Alotaibi, Internet of Things security: a survey, J. Netw. Comput. Appl. 88 (2017) 10–28, doi:10.1016/j.jnca.2017.04.002.
[21] A. Riahi Sfar, E. Natalizio, Y. Challal, Z. Chtourou, A roadmap for security challenges in the Internet of Things, Dig. Commun. Netw. 4 (2) (2018) 118–137, doi:10.1016/j.dcan.2017.04.003.
[22] S. Sicari, A. Rizzardi, L. Grieco, A. Coen-Porisini, Security, privacy and trust in Internet of Things: The road ahead, Comput. Netw. 76 (2015) 146–164.
[23] R. Roman, J. Zhou, J. Lopez, On the features and challenges of security and privacy in distributed internet of things, Comput. Netw. 57 (10) (2013) 2266–2279, doi:10.1016/j.comnet.2012.12.018.
[24] P. Kasinathan, C. Pastrone, M.A. Spirito, M. Vinkovits, Denial-of-Service detection in 6LoWPAN based Internet of Things, in: Proceedings of the International Conference on Wireless and Mobile Computing, Networking and Communications, 2013, pp. 600–607, doi:10.1109/WiMOB.2013.6673419.
[25] E. Hodo, X. Bellekens, A. Hamilton, P. Dubouilh, E. Iorkyase, C. Tachtatzis, R. Atkinson, Threat analysis of iot networks using artificial neural network intrusion detection system, in: Proceedings of the 2016 International Symposium on Networks, Computers and Communications (ISNCC), 2016, pp. 1–6, doi:10.1109/ISNCC.2016.7746067.
[26] M. Lewis, Using graph databases to assess the security of thingernets based on the thingabilities and thingertivity of things, in: Proceedings of the Living in the Internet of Things: Cybersecurity of the IoT – 2018, Institution of Engineering and Technology, 2018, pp. 1–9, doi:10.1049/cp.2018.0008.
[27] National Institute of Standards and Technology, SP 800–53 Rev 4: Recommended Security and Privacy Controls for Federal Information Systems and Organizations, Technical Report, 2013.
[28] MicroBees, The MicroBees web site, 2018.
[29] M. Rak, Security assurance of (multi-)cloud application with security SLA composition, Lect. Notes Comput. Sci. 10232 (2017) 786–799.
[30] V. Casola, A. De Benedictis, M. Rak, U. Villano, Security-by-design in multi-cloud applications: an optimization approach, Inf. Sci. 454–455 (2018) 344–362, doi:10.1016/j.ins.2018.04.081.
[31] Microsoft Corporation, The STRIDE Threat Model, 2016 https://docs.microsoft.com/en-us/previous-versions/commerce-server/ee823878(v=cs.20).
[32] V. Casola, A. De Benedictis, M. Rak, U. Villano, Towards automated penetration testing for cloud applications, in: Proceedings of the 2018 IEEE Twenty-seventh International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2018, pp. 24–29, doi:10.1109/WETICE.2018.00012.