Metadata of the chapter that will be visualized online

Chapter Title	IoT Security Monitoring Tools and Models	
Copyright Year	2024	
Copyright Holder	The Author(s), under exclusive license to Springer Nature Switzerland AG	
Corresponding Author	Family Name	**Cavalli**
	Particle	
	Given Name	**Ana Rosa**
	Suffix	
	Division	Réseaux et Services Mobiles
	Organization	Institute Polytechnique de Paris
	Address	Evry, France
	Email	ana.cavalli@telecom-sudparis.eu
Corresponding Author	Family Name	**Mallouli**
	Particle	
	Given Name	**Wissam**
	Suffix	
	Organization	Montimage
	Address	Paris, France
	Email	wissam.mallouli@montimage.com
Corresponding Author	Family Name	**de Oca**
	Particle	
	Given Name	**Edgardo Montes**
	Suffix	
	Organization	Montimage
	Address	Paris, France
	Email	Edgardo.montesdeoca@montimage.com
Author	Family Name	**Rivera**
	Particle	
	Given Name	**Diego**
	Suffix	
	Organization	Montimage
	Address	Paris, France
	Email	diego.rivera@montimage.com

13

Abstract

Systems based on Internet of Things are more and more being used in many different critical domains. However, such devices introduce new vulnerabilities and trying to cope with them on devices that have limited resources remains a challenge. AU3

This chapter presents ongoing research on security modeling and monitoring for IoT networks and describes some of the most popular tools used. First, it describes different reference models and architectures designed for IoT and highlights the need for security features to provide the required trust and privacy. A presentation of the different security models developed by different research teams allows identifying the most salient features and the challenges that still remain to be answered.

This chapter also details different attack types that have high impact on IoT networks, some that are common in the Internet but others that are more specific to IoT. Finally, the chapter provides a description of a concrete example on how network monitoring and security analysis can be used to detect anomalous and malicious behavior.

This chapter concludes by identifying some of the problems that still need to be addressed.

Keywords (separated by '-')	Internet of Things - Security - Monitoring - Security architectures - Attacks - Detection - Security models - Security tools

13 IoT Security Monitoring Tools and Models

Ana Rosa Cavalli, Wissam Mallouli, Edgardo Montes de Oca, and Diego Rivera

Contents

A. R. Cavalli (✉)
Réseaux et Services Mobiles, Institute Polytechnique de Paris, Evry, France
e-mail: ana.cavalli@telecom-sudparis.eu

W. Mallouli (✉) · E. M. de Oca (✉) · D. Rivera
Montimage, Paris, France
e-mail: wissam.mallouli@montimage.com; Edgardo.montesdeoca@montimage.com; diego.rivera@montimage.com

Abstract

Systems based on Internet of Things are more and more being used in many different critical domains. However, such devices introduce new vulnerabilities and trying to cope with them on devices that have limited resources remains a challenge.

This chapter presents ongoing research on security modeling and monitoring for IoT networks and describes some of the most popular tools used. First, it describes different reference models and architectures designed for IoT and highlights the need for security features to provide the required trust and privacy. A presentation of the different security models developed by different research teams allows identifying the most salient features and the challenges that still remain to be answered.

This chapter also details different attack types that have high impact on IoT networks, some that are common in the Internet but others that are more specific to IoT. Finally, the chapter provides a description of a concrete example on how network monitoring and security analysis can be used to detect anomalous and malicious behavior.

This chapter concludes by identifying some of the problems that still need to be addressed.

Keywords

Internet of Things · Security · Monitoring · Security architectures · Attacks · Detection · Security models · Security tools

13.1 Introduction

Internet connectivity in Internet of things (IoT) systems has become a ubiquitous service, being used more and more in industrial and critical systems, but also in the city (e.g., managing traffic lights, air sensors), and even in homes (e.g., managing intelligent lightning, heating and energy

S. Ziegler et al. (eds.), *Springer Handbook of Internet of Things*, Springer Handbooks,
https://doi.org/10.1007/978-3-031-39650-2_13

consumption, intelligent locking systems for the doors). However, bringing connectivity to such devices introduces security vulnerabilities and concerns in the IoT networks, making them a target for many different attacks. The challenge arises when trying to cope with such security issues on devices that have limitations in electrical consumption and computational power. Despite the fact that monitoring techniques already exist for traditional networks, there are two principal limitations that do not allow directly applying them on next-generation IoT networks. On the one hand, on-the-fly security analysis requires the processing of big amounts of data that need sufficient computational power; hence, it cannot be performed on site [1]. On the other hand, most of the network security solutions (e.g., firewalls, Intrusion Detection and Prevention Systems) have been conceived to work on the edge or the limits of the network to protect the network from external attacks and are mainly based on Internet Protocol network traffic. Since IoT networks do not have a clear border, it becomes easy for an attacker to insert a new device in the network and infect it from the inside. These observations tend to show that security analysis needs to be reformulated to consider the restrictions on embedded devices and the new inherent vulnerabilities. Research is striving to solve these challenges by introducing new concepts and techniques, in particular virtualization techniques. New tools are also appearing in the market targeting to improve the security of IoT networks.

IoT is a concept that describes a network of interconnected devices capable of interacting with other devices, human beings and its surrounding physical world to perform a variety of tasks [2]. Modern IoT devices make use of sensors (e.g., accelerometer, gyroscope, microphone, light sensor, etc.) [3] to detect any changes in their surrounding and take necessary actions to improve any ongoing task efficiently [4]. The increasing popularity and utility of IoT devices in different application domains are stimulating the growth of IoT industry at a tremendous rate. According to a report by Business Insider [5], 30 billion devices will be connected to the Internet by 2020.

These devices can provide new functionality in different domains, but can also be used as vehicles to launch attacks (examples can be found for instance in [6–11]).

The challenge of **security monitoring** on IoT network arises when trying to detect these attacks on devices that have strict resource limitations. Existing centralized monitoring techniques (Intrusion Detection and Prevention Systems) cannot handle the large amounts of data that needs to be analyzed and have been designed to work on the edge of the networks and cannot cope with IoT networks that lack clear boundaries. Furthermore, depending on the application domain, security monitoring can be done to detect anomalies by analyzing the protocol/message exchanges; or, in the case of time series measurements, the statistics and trends of the measured values.

In the following sections, we present some of the ongoing research on security modeling and monitoring for IoT networks, and present some of the most popular tools. Finally, we provide a conclusion that identifies the problems that still need to be addressed.

13.2 Models for IoT Systems

13.2.1 Reference Models

Networks, computations, applications and data management architectures that are IoT compatible require a different communication and processing model. In [12], the authors argue that a new **reference model** is needed for IoT systems. They stress the fact that a standard way of “*understanding or describing these models for the IoT*” is missing. The consequence is that there is some confusion between what is an IoT device and application and what is not. However, when data are “*generated under the control of machines or equipment and sent across a network, it is probably an IoT system.*” Cisco proposes an IoT Reference Model that is composed of seven levels. The objective is to provide clear definitions and specifications that give a precise definition of the elements and functions of IoT systems and applications.

Table 13.1 represents the different levels of the IoT Reference model that can be described as follows:

1. Level 1 represents the “things,” which are physical devices and controllers that might control multiple devices. Each can send and receive information. As mentioned in [12], the “*devices are diverse, and there are no rules about size, location, form factor, or origin. Some devices will be the size of a silicon chip. Some will be as large as vehicles.”*
2. Level 2 represents communications and connectivity. It includes information transmission between devices (Level 1) and the network, across the network and between the networks (Level 2), and low-level information processing occurring at Level 3. Level 2 includes reliable delivery across the networks; switching and routing and security at the network level.
3. Level 3 corresponds to Edge Computing and represents the conversion of network data flows into information that is suitable for storage and higher-level processing at Level 4 (data accumulation). This means that Level 3 activities focus on high-volume data analysis and transformation. The information processing is as close to the edge of the network as possible and is often called Fog Computing.
4. Level 4 corresponds to Data Accumulation and determines what data are interesting for the higher levels and

Table 13.1 The seven levels of the IoT reference model

Situation with respect to the network	Level	Name	Main characteristics	Type of security	Type impacted
Center	7	Collaboration and Processes	People and business processes	Identity management	Software
	6	Application	Analytics, reporting and control	Authentication/ Authorization	Software
	5	Data abstraction	Data aggregation and access	Secure storage	Hardware and software
	4	Data accumulation	Data storage	Tamper resistant	Software
	3	Edge (fog) computing	Data element analysis and transformation	Secure communications	Protocols and encryption
	2	Connectivity	Communication and processing units	Secure network access	Hardware and protocols
Edge	1	“Things”	Physical devices and controllers	Secure content	Silicon

implements the needs for persistence, organization, combination, recomputation, aggregation and storage type.

5. Level 5 corresponds to Data Abstraction and is focused “*on rendering data and its storage in ways that enable developing simpler, performance-enhanced applications.*” Level 5 is assumed to process different things, as for instance: “*reconciling multiple data formats from different sources,*” “*assuring consistent semantics of data across sources,*” “*confirming that data is complete to the higher-level application*” and “*protecting data with appropriate authentication and authorization*” mechanisms.
6. Level 6 is the Application level. The Reference Model does “*not strictly define an application.*” Applications are diverse and “*based on vertical markets, the nature of device data, and business needs.*” As examples can be mentioned: Control Applications, Vertical and Mobile Applications, and Business Intelligence and Analytics applications.
7. Level 7 corresponds to Collaboration and Processes. This level involves people and business processes that are involved in the IoT system and its functions or services. The IoT system creates information that is “*of little value*” unless it produces actions, with participation of people and processes.

The focus of security at each level is given in the “Type of Security” column. Formal modeling techniques can be used in IoT for specifying and analyzing functional correctness, attack scenarios, verifying security and privacy properties, and specifying corrective actions.

[13] identifies the different reference models for IoT that have or are being developed by standardization bodies, projects and associations. The industrial sector is the main driving force for the standardization that is deemed necessary to “facilitate interoperability, simplify development, and ease implementation.”

Table 13.2 gives an overview of the different initiatives.

In all these reference models, “security features are necessary to provide trust and privacy and are required for all aspects of the IoT.”

As can be seen, Industrial IoT (IIoT) is the major driving force for the standardization of IoT for the manufacturing sector [14]. provides an analysis of existing IIoT reference frameworks, comparing them and identifying gaps. The authors identify cyber security as one of the major trends considered by most reference architectures. All address security and trust-related concerns but the scope is usually limited to high-level descriptions with little concrete specifications and recommendations. IIoT makes isolation of critical infrastructure and devices behind restrictive firewalls practically impossible. Furthermore, besides isolation and cyber threat detection/mitigation/prevention, many other aspects need to be covered that include certification processes, provenance tracking, network security and process isolation. These need to be regarded at all levels and from an end-to-end perspective starting with embedded devices, edge/fog/cloud computing, highly distributed systems and application domains. Articles, such as [15], present the security and privacy issues in IIoT, advocating a holistic security framework and network-wide detection of intrusion attempts. But, which method or technology should be used at what position of the architecture is mostly missing.

13.2.2 Models for Security

Several research efforts have been undertaken concerning IoT **modeling formalisms** and, in particular, some deal with security. Some examples are given in the following paragraphs.

One of the techniques used is **attack trees** to specify possible attacks as done in [16]. The authors mention that some recent research activities regarding formal modeling and correctness analysis of IoT systems present limitations, t2.1 **Table 13.2** IoT reference model initiatives

	Initiative	Main related aspects	Links
t2.2	Initiative	Main related aspects	Links
t2.3	Reference Architecture Model Industry 4.0 (RAMI 4.0)	Concerned with the standardization of IoT for smart factories. It goes beyond the IoT by adding manufacturing and logistics details. It is domain specific, dealing with the life cycle and value streams of manufacturing applications. Security requirements are identified and outlined in chap. 7 of the Implementation Strategy for Industry 4.0. In order to keep the core functionality in a factory free from faults, even when the "external" network is experiencing attacks, requirements on "separability" (in other words, isolation) and "security by design" of the infrastructure are defined. This does not go further than high-level specification of requirements.	https://www.zvei.org/en/subjects/industrie-4-0/the-reference-architectural-model-rami-40-and-the-industrie-40-component/
t2.4	Industrial Internet Reference Architecture (IIRA)	Has a strong industrial focus and provides a detailed view of the IoT's information technology aspects. It focuses on the "functionality of the industry domain, such as business, operations (prognostics, monitoring, optimization and so on), information (analytics and data) and application (User Interfaces (UI), Application Programming Interfaces (API), logic and rules)." A security framework has been designed aiming at identifying and positioning security-related architectures, designs and technologies, as well as identifying procedures relevant to trustworthy **Industrial IoT** (IIoT) systems. It considers both Operational and Information Technology aspects and the differences that impact security. It also identifies the building blocks for the framework: End-point Protection, Communications and Connectivity Protection, Security Monitoring and Analysis, Security Configuration and Management. Security Monitoring implies a Monitor-Analyze-Act cycle (that may be in real-time or not) to identify usage patterns and detect/mitigate/prevent potential attack scenarios. The Monitor function must capture data from the end-points and communications, the secure remote logging and supply chain. The Analyze function needs to consider behavior and rule-based analysis. The Act function includes Proactive/Predictive mitigation, Reactive detection & Recovery and Root Cause/Forensics. The functional specification is more complete and some of the architectural issues and guidelines are covered.	https://www.iiconsortium.org/pdf/IIC_PUB_G4_V1.00_PB-3.pdf
	IoT-Architecture (IoT-A)	Provides a detailed architecture and model including functional and information perspectives, and system requirements. It "concentrates on the generic aspects of informatics instead of the application facets of semantics." The IoT Reference Model and Architecture is described in detail and	https://www.researchgate.net/publication/272814818_Internet_of_Things_-_Architecture_IoT-A_Deliverable_D15_-_Final_architectural_reference_model_for_the_IoT_v30

(continued)

Table 13.2 (continued)

Initiative	Main related aspects	Links
	guidelines are given. A Security Functionality Group is defined for ensuring the security and privacy of IoT-A-compliant systems. It is in charge of handling the access of a client to the system, protecting private parameters of users based on anonymity, ensuring that legitimate interaction occurs between peers based on authorization functions or through the reliance on a trust-and-reputation model, and enabling secure communications between peers by managing the establishment of integrity and confidentiality features. Thus, it consists of five functional components: Authorization; Key Exchange & Management; Trust & Reputation; Identity Management; and Authentication. Other security aspects and requirements are discussed but remain high level.	
Standard for an Architectural Framework for the IoT	IEEE P2413 project working group focusing on IoT architectural framework and, in particular, addressing protection, security, privacy and safety issues. It targets implementations leveraging cross-domain interaction and semantic interoperability among various domains and components of a Smart City. This standard leverages the architectural framework for IoT defined in the draft of IEEE P2413 standard, which relies on the international standard ISO/IEC/IEEE 42010.	https://standards.ieee.org/develop/project/2413.html
Arrowhead Framework	Focuses on interoperability of embedded devices and cooperative automation.	www.arrowhead.eu
Other initiatives related to Machine-to-Machine (M2M)	ETSI TC and ITU-T Machine-to-Machine (M2M) standards are also closely related to IoT. Specifications for a standardized platform: include: Requirements (ETSI TS 102689), Functional architecture (ETSI TS 102690) and Interface descriptions (ETSI TS 102921). M2M The security framework lays down the underlying functions and key hierarchy pertaining to M2M security, addresses the bootstrapping and service provisioning of D/G M2M Nodes, describes the security procedures for M2M Service Connection between the Device/Gateway M2M Node and the Network Domain, and addresses the security of the *mId* (M2M to device interface) used for the inter-Service Capability Layer communications.	https://www.etsi.org/technologies/internet-of-things

(continued)

as for instance, to produce abstract behavioral patterns and modeling attacks following these patterns, which limits the possibility to describe richer behaviors. They propose IoT-SEC, a framework that defines an adequate semantics for the IoT's components and their interactions. This model includes social actors that behave differently than automated processes. For security analysis, they develop an approach based on attack trees from where they automatically generate the monitor, the security policies and requirements to reinforce the IoT model and to be able to verify that the model is secure.

The IoT-SEC framework introduces a modeling formalism that captures the underlying semantics of IoT. The formalism is rich enough "*to cover social behaviors, physical and digital objects, communication protocols, internal and external servers, and computation and storing cloud services.*" IoT-SEC also models a library of intruders that are particular processes for each IoT components acting maliciously. Regarding security, they develop a security analysis methodology for IoT, which relies on statistical analysis and model-checking approaches based on the PRISM tool [17]. This tool is used to verify the functionality and to check the security properties of the IoT model.

In order to automate the application of the IoT-SEC tools, the authors have defined a mapping from the IoT models, expressed in the proposed formalism, to the PRISM formalism. To overcome the limitations of PRISM regarding the expressiveness of monitors and security properties, the authors propose "*a library of pre-configured attack trees and develop instantiation mechanisms that help to generate automatically relevant monitors and security properties.*"

PRISM "*is a probabilistic symbolic model checker that checks probabilistic specifications over probabilistic models*" [17]. The specifications can be described either using probabilistic computation tree (PCTL) [18] or stochastic logic. The PRISM language is used to specify a model and program a set of modules, each having local Boolean or integer variables. A module's state is defined by the values of its local variables, and the program's state by the evaluation of all variables, local and global. The behavior of a module is defined by a set of probabilistic commands that specifies the effect of an action in a probabilistic transition system.

With respect to other work, IoT-SEC covers the probability and costs of actions, formalizes IoT, analyzes the correctness and measures their security level. Moreover, IoT-SEC allows automation based on probabilistic model checking.

Another attack tree approach is described in [19] that proposes a **modeling language** for the security of IoT systems that represents data and access controls. The language permits the users to create the models of their IoT systems and analyze the probability of cyber-attacks occurring and succeeding. The modeling language allows describing interactions between human actuators and/or things that could be hardware, sensors, software tools, etc. The human behavior is a key element for the security analysis and can be unintentionally or maliciously harmful. The security failures are modeled following the attack tree approach. The modeling language is transformed to a component-based model called BIP [20] for performing security analyses and applying formal verification techniques developed, for instance, to detect deadlocks. The authors proved the correctness of the proposed transformation, implemented it and illustrated the application of the technique to a case study involving cyber-attacks on a smart hospital.

Other research works propose formal techniques such as satisfiability solvers, provers and color Petri nets. Different notable examples are:

- A security analysis approach proposed by [21] based on the SMT (**Satisfiability Module Theory**) solver for IoT entities. It is focused on device configurations, network topologies, user policies and their related attack surfaces. Entities are described as high-order logic formulas, and the policies are described as a set of discrete constraints. In order to verify existing vulnerabilities, SMT solver outputs the possible solutions satisfying the constraints within an attack formula. The proposed approach is limited to strict IoT schemes and the analysis method is not automated.
- A formal approach is investigated by [22] that shows how the Isabelle **prover** [23] can help improve detection of attacks in traces of IoT e-health systems by combining "*ethical requirement elicitation with automated reasoning.*" In order to provide trustworthy and secure IoT environments in health-care scenarios, the authors employ high-level logical modeling using dedicated Isabelle frameworks for describing: infrastructures, human actors, security policies, attack tree analysis and security protocols.
- To achieve high-level instantiation of the run-time verification, [24] uses color **Petri nets**. The authors integrate runtime verification enablers in the feedback adaptation loop to guarantee the achievement of self-adaptive security and privacy properties for e-health settings. At run-time, the authors enable the contextual state model, the requirements specifications and the dynamic context monitoring and adaptation.

More holistic approaches involve defining interactions and roles, and defining security management requirements and mechanisms:

- The security mechanisms design and deployment for IoT presented by [25] introduces a new paradigm of security, which "*consider the security problem from a holistic perspective including the new actors and their interactions.*" The authors propose a systemic approach for IoT security that is presented in the thesis of one of the authors [26]. The model comprises four nodes: person, technological ecosystem, process and intelligent object. The last node is the newest and reflects the IoT dimension. These nodes interact through tensions, namely: identification, trust, privacy, safety, auto-immunity, reliability and responsibility. The authors aimed at defining each node and its roles, describing each tension's meaning, effect, challenges that need to be addressed, and applied them to real examples taken from classical application domains to substantiate the use of systemic approaches.

- An even wider approach is proposed by [27] that first presents a thorough overview on the introduction of IoT including history, components, connection and application of IoT, and then proposes an IoT layer architecture: namely, the coding, perception, network, middleware, application and business layers. The authors represent a more developer-oriented viewpoint that maps only partially to the layers presented in Table 13.1 that are more data-oriented [27]. also presents IoT security and privacy requirements and challenges, types and targets of attacks to detect and prevent. A model is proposed that targets supporting the security management for IoT. It incorporates the appropriate security mechanisms and protocols for the different IoT security layers.

Researchers also introduce techniques that improve the management of security and reliability, for instance based on **virtualization**, **adaptation** and **cognitive techniques**:

- An architecture that separates the physical and virtual instances of sensors, gateways, application servers and data storage is proposed by [28]. In this way, virtualized sensor nodes can more easily be guaranteed to assure security, privacy, reliability and data protection, including secure association, authentication and authorization, privacy, data integrity and protection. The authors indicate that the only bottleneck is the physical interaction between the real sensor and its virtual counterpart. Nevertheless, latency will also be impacted depending on the type of security analysis and management that is applied.
- To improve the adaptability of IoT systems at runtime, [29] proposes a model-driven approach. They authors realize adaptive IoT systems by facilitating the modeling through an extension of SysML [30] called SysML4IoT [31]. This allows specifying both functions and adaptations. The authors first defined the system requirements by creating a design model that captures the system functionality and its adaptation. The functionality is modeled by the SysML4IoT profile, while the environment and the interactions with the system is modeled following a publish/subscribe paradigm. A state machine approach is used to model the runtime adaptation. From the model, the authors generate the system implementations by transforming the high-level design model to an IoT platform specific model. This is then used to generate the Java code. This generated code is deployed on the hardware platform of the system and a smart lighting system use case allows validating the results.
- Finally, to reduce system design complexity, [32] proposes a set of design patterns. The goal is to manage the context changeability at runtime by introducing autonomic cognitive management patterns that identify a combination of management processes able to continuously detect and manage the context changes. A healthcare use case, the patient comorbidity management based on wearables, is used to validate the results. The authors propose four maturity levels that define the different stages that an IoT-based system implements to reach the smart manageability. For each level, they define a design pattern that integrates a set of autonomic and cognitive capabilities which are selected based on the system requirements. The most mature, Autonomic Cognitive Management, pattern is described to manage the context changeability and coordinate the business processes based on the collected data from IoT.

13.3 IoT-Layered Monitoring Architectures

There is no single and general agreement about a monitoring architecture for IoT-based environments. Many and different architectures have been proposed by researchers and experts in the literature [2, 3, 5, 8, 9, 33, 34]. According to some researchers, IoT monitoring architecture has three layers; others support four or even five-layer architecture where requirements of IoT regarding security and privacy can be fulfilled.

The hierarchies of the proposed **layered architectures** of Internet of Things (IoT) are shown in Fig. 13.1. All contain the perception, network/transport and application layers. In the four-layer architecture, a new paradigm is presented and is called the support layer. In the five-layer architecture, the concepts of processing and business layers are introduced. More details about each layer and the potential security issues to be monitored in each are presented in the next subsections.

13.3.1 Three-Layer Architecture

The **three-layer architecture** is the very basic monitoring architecture that fulfills the main concepts of IoT. It was proposed in the early stages of development of IoT [5, 8, 33] environments. It has three layers named Perception, Network and Application as shown in Fig. 13.2. These are detailed in the following paragraphs.

Perception Layer

The **Perception layer** is also known as a Sensor layer. It works like a person's eyes, ears and nose. It has the responsibility of identifying things and collecting information from them. There are many types of sensors attached to objects for

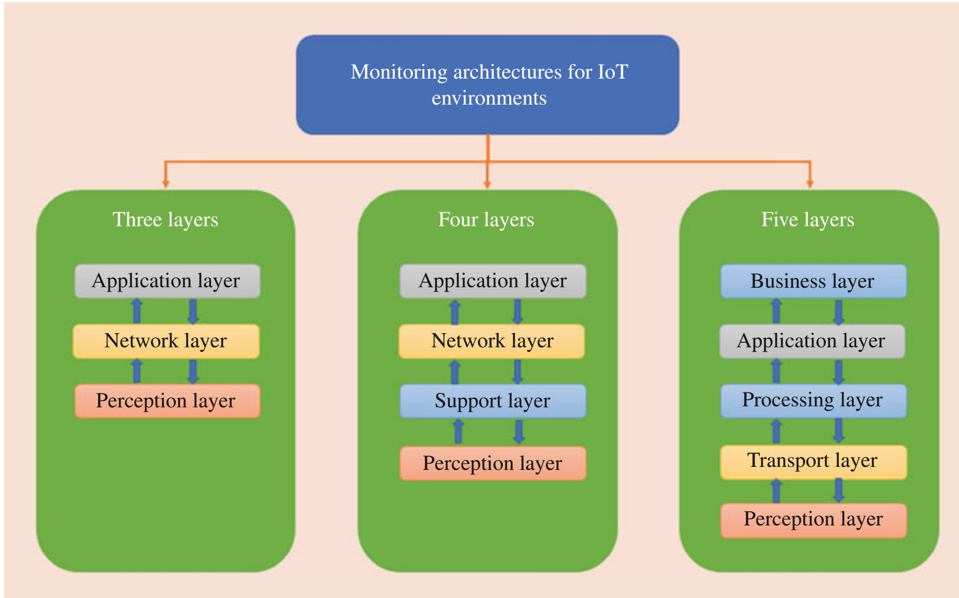

Fig. 13.1 The layered architectures of IoT (three, four and five layers)

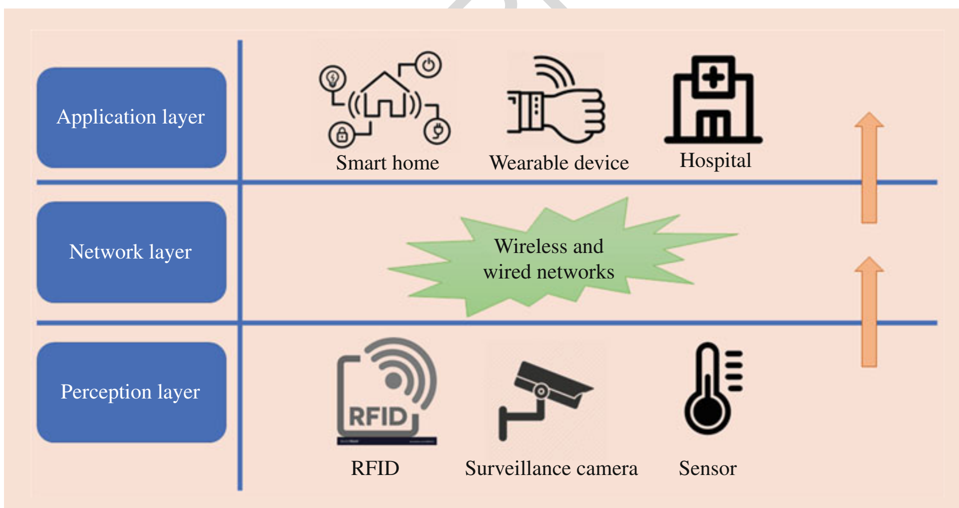

Fig. 13.2 The three-layered monitoring architecture of IoT

collecting information such as Radio Frequency Identification tags (RFID), 2-dimensional barcode, sensors, etc. The sensors are chosen according to the requirements of the applications. The information that is collected by these sensors can be about location, changes in the air, environment, motion, vibration, temperature, etc. They can be the main target of attackers who wish to utilize them to change the sensor outcomes with their own. Thus, the majority of threats are related to the sensors themselves [3, 9, 34]. Common security threats of the Perception layer are:

- **Eavesdropping**: Eavesdropping is an unauthorized real-time attack where private communications, such as phone calls, text messages, fax transmissions or video conferences are intercepted by an attacker. The intention is to steal information that is transmitted over a network. It takes advantage of insecure transmissions when accessing information being sent and received.
- **Node Capture**: It is one of the hazardous attacks faced in the Perception layer of IoT. Here, an attacker gains full control over a key node, such as a gateway node. It may leak all the information, including the communications between the senders and the receivers, and the keys used to make secure communications and data storing [35].
- **Fake Malicious Node**: These types of attacks correspond to an attacker that adds a node to the system and inputs

fake data. It aims at disrupting the transmission of real information. Furthermore, a node added by an attacker can provoke the consumption of the limited energy of real nodes and potentially gain control in order to destroy or severely disrupt the network.

- **Replay Attack**: It is also known as a play back attack. It is an attack in which an intruder eavesdrops on the conversation between a sender and receiver, and captures authentic information from the sender. In this way, the intruder can send the same authenticated information to the victim that had already been received by it, showing proof of its identity and authenticity. The message is in encrypted form, so the receiver may treat it as a correct request and take action, provoking undesired behavior or consuming energy as desired by the intruder [7].
- **Timing Attack**: It is usually used in devices that have weak computing capabilities. It enables an attacker to discover vulnerabilities and extract secrets maintained in the security of a system by observing how long the system takes to respond to different queries, input or cryptographic algorithms [10].

The monitoring of this layer means that we monitor the real devices that are deployed in the IoT environment. A nonauthorized device or a repeated message should be seen as vulnerable.

Network Layer

The **Network layer** is also known as the Transmission layer. It acts like a bridge between the Perception layer and Application layer. It carries and transmits the information collected from the physical objects through the sensors. The medium for the transmission can be wireless or wire based. It also takes the responsibility for connecting the smart things, network devices and networks to each other. Therefore, it is highly sensitive to attacks. It has prominent security issues regarding integrity and authentication of information that is being transported in the network. Common security threats and problems in the Network layer are:

- **Denial of Service** (DoS) Attack: A Denial of Service attack is an attack to prevent authentic users from accessing devices or other network resources. It is typically accomplished by flooding the targeted devices or network resources with redundant requests in order to make it impossible or difficult for some or all of the authentic users to use them [11].
- **Man-in-The-Middle** (MiTM) Attack: The Man-in-The-Middle attack is an attack where the attacker secretly intercepts and alters the communication between a sender and a receiver who believe they are directly communicating with each other. Since an attacker controls the communication, he or she can change messages according to their objectives. It can cause serious threats to online security because they give the attacker the facility to capture and manipulate information in real time [36].
- **Storage Attack**: The information of users is stored on storage devices or in the cloud. Both storage devices and cloud can be attacked by the attacker. In this way, the user's information may be compromised or changed to incorrect values. The replication of information associated with the access of other information by different types of persons provides more opportunities for attacks.
- **Exploit Attack**: An exploit is any immoral or illegal attack in the form of software, chunks of data or sequences of commands. It takes advantage of security vulnerabilities in an application, system or hardware. It usually comes with the aim of gaining control of the system and stealing information stored in a network [6].

The monitoring of the Network layer means that we have the possibility to capture and decode transmitted packets. It is then possible to analyze them to detect anomalies, misbehaviors and attacks. Notice that the protocols used by IoT devices are proprietary (ZigBee, 6LowPAN, CoAP, etc.) and can run directly over the Ethernet layer, meaning that no IP layer is provided. This constitutes a real challenge that IP-based Intrusion Detection Systems do not address. Furthermore, the capturing of the communications needs to be done on the wireless part since many cyber-attacks are not observable from the Internet traffic after the gateway or bridge.

Application Layer

The **Application layer** comprises all the applications that use the IoT technology or for which IoT has been deployed. The applications of IoT can concern different domains such as smart homes, smart cities, smart health, animal tracking, etc. The applications have the responsibility of providing the services to the users. The services may vary for each application because services depend on the information that is collected by the sensors. In the Application layer, security is one of the key issues. In particular, when IoT is used in order to provide a smart home system, it introduces many threats and vulnerabilities both from the inside and outside of the system. To implement strong security in an IoT-based smart home, one of the main constrains is that the devices used have weak computational power and a low amount of storage such as ZigBee [37]. Common security threats and problem in the Application layer are:

- **Cross Site Scripting**: This is an injection attack. It enables an attacker to insert a client-side script, such as java script in a trusted site viewed other users. By doing so, an attacker can completely change the contents of the application according to his or her needs and use the original information in an illegal way [4].

- **Malicious Code Attack**: This corresponds to some code in any part of the software of the system that has the intention to cause undesired effects and damage to the system. It is a type of threat that may not be blocked or controlled by the use of antivirus tools. It can either activate itself or act as part of the program requiring the user's attention to perform an action.
- **Massive Data and Processing**: Due to the large number of devices and the massive amounts of data transmitted between users, it can occur that it becomes difficult or impossible to deal with the data processing needed as defined by the requirements. This risk can be increased by attackers that provoke the generation of more data and processing similar to Denial of Service attacks. As a result, this can lead to network and service disturbance and data loss.

Monitoring the Application layer can be done by classical means used in **Business Application Monitoring** (BAM) based on analyzing the packet payloads, for example, for malware detection, and analyzing application logs, for example, using Security Information Management Systems (SIEM).

13.3.2 Four-Layer Architecture

Due to the continued development in IoT, researchers have proposed secure monitoring based on a **four-layer architecture** [38]. This architecture has the same three layers like the previous architecture, but with Support layer added. Figure 13.3 presents the four-layered architecture along with the recommended security mechanisms used to make it secure from intruders. The three layers have the same functionality as the Three-layer architecture that we have already discussed previously so the functionality of the Support layer with respect to security attacks is as explained in the following paragraphs.

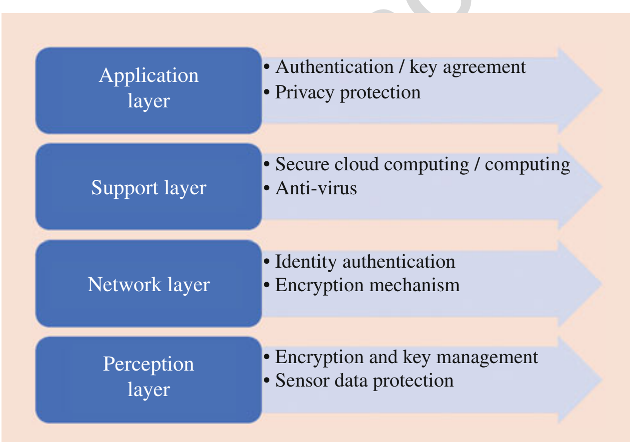

Fig. 13.3 The four-layered architecture of IoT along recommended security mechanisms

Support Layer

The reason for introducing a fourth layer is for improving the security-by-design characteristics in the architecture of IoT. In a three-layer architecture, information is sent directly to the Network layer. Sending information directly to the Network layer increases the possibilities of attacks. In the four-layer architecture, information is sent to a Support layer coming from the Perception layer. The **Support layer** has two responsibilities:

- First, it confirms that the information is sent by the authorized users and does not contain any threats. There are many ways to verify the users and the information. The most commonly used method to verify the users is by authentication. It is implemented by using preshared secrets, keys and passwords. The most common way to detect malware is using antivirus and malicious scanning software.
- Second, the Support layer is responsible for sending the information to the Network layer. The medium to transmit information from the Support layer to Network layer can be wireless or wire-based and secured more thoroughly using different techniques such as encrypting or obfuscating the information.

There are various attacks that can affect this layer such as Denial of Services attacks, malicious insider attacks, unauthorized accesses, etc. Common threats and problems of the Support layer are:

- Denial of Service Attack: The Denial of Service attack in a support layer is related to the network layer. An attacker sends a large amount of data to flood the network traffic. This leads to the massive consumption of the system's resources and the exhausting the IoT devices, and makes the user not capable of accessing the system or services.
- Malicious Insider Attack: It occurs from the inside of an IoT network environment with the objective of accessing the personal information of the users. It is performed by an authorized user obtaining access the information of other users. It is an attack that is sometimes complicated and difficult to detect, and requires different mechanisms to prevent the threat [39, 40].

13.3.3 Five-Layer Architecture

The four-layer architecture played an important role in the development of IoT. There were some issues regarding security and storage in this architecture. To remediate them, researchers proposed a **five-layer architecture** to make the IoT even more secure [41–43]. It has the three layers as in the case of the previous architectures: Perception, Transport and Application layers. It also has two more layers. The names of these newly proposed layers are the Processing layer and Business layer. It is considered that this newly proposed architecture has the ability to fulfill all of the requirements of IoT. It also has the ability to make the applications of IoT more secure. The workings of these layers and security attacks that can affect them are detailed in the following paragraphs.

Processing Layer

The **Processing layer** is also known as a Middleware layer. It collects the information that is sent from the Transport layer. It performs the processing of the collected information. It has the responsibility of eliminating extra information that has no meaning and extracting the useful information. However, it also removes the problem of dealing with big data in IoT. In big data, a large amount of information is received which can affect the performance of the IoT functions and services. There are numerous attacks that can affect the Processing layer and disturb the performance of the IoT system. Common attacks are:

- **Exhaustion**: An attacker uses exhaustion to disturb the processing of the IoT system. It occurs as an after-effect of attacks, such as Denial of Service attacks, in which an attacker sends the victim many requests to make the network unavailable for them. It could be a result of other attacks that aim at exhausting the system resources, such as the battery and memory resources [44].
- **Malwares**: This is an attack on the confidentiality of the information of users. It refers to the exploitation of Viruses, Spyware, Adware, Trojans horses and Worms that act to disrupt or change the behavior of the system. It takes the form of executable codes, scripts and contents. It acts against the requirements of system and compromise the confidentially of information [45].

Business Layer

The **Business layer** concerns the intended behavior of an application and acts like a manager of the whole system. It has the responsibility to manage and control the application, business and profit models of IoT system. The user's privacy is also managed by this layer. It has the ability to determine how information can be created, stored and changed. Vulnerability in this layer permits the attackers to misuse an application by interfering on the business logic. Most problems regarding the security of this layer concern the weaknesses in an application that result from a broken, vulnerable or missing security control. Common problems regarding security of the Business layer are:

- **Business Logic Attack**: This attack takes advantage of a flaw in a program. This flaw allows it to obtain control and affect the exchanges of information between a user and a supporting database of an application. There are several common flaws in the business layer, such as improper coding by a programmer, incorrect password recovery and validation, incorrect input validation and vulnerable encryption techniques [46].
- **Zero-Day Attack**: This refers to a security hole or a problem in an application that has not yet been identified by the vendors or the security community. This security hole is exploited by the attacker to take control of the system without the user's consent and without their knowledge [47, 48].

13.4 Security Modeling Tools

Security modeling can concern several aspects as, for instance:

- Modeling the IoT system during the different phases of development for improving it resiliency and eliminating vulnerabilities. This includes the introduction of Domain-Specific Languages (DSLs).
- Modeling the system for performing simulations.
- Threat modeling to be able to validate the resiliency of the system to attacks (in other words, penetration or attack testing).

The research community proposes a number of IoT related tools to support specific methodologies and frameworks. In Table 13.3 are presented several examples that introduce **Domain-Specific Languages**, extraction of metadata from models, simulators:

13.5 Research on Monitoring of IoT Environments

13.5.1 IoT-Tailored Security Monitoring Tools

Despite the fact that existing **monitoring tools** are not designed to work in IoT environments, a first approach is trying to adapt the existing tools to make them work on the

Table 13.3 Security modeling tools for IoT

Type	Examples
DSLs	ASTo (Apparatus Software Tool) is proposed in [49]. It is a software tool for security analysis of IoT systems that allows visualizing IoT systems using a domain-specific modeling language. The modeling language allows expressing hardware, software and social aspects, as well as security concepts. Two metamodels are used to describe IoT systems: (1) for the design phase to identify the assets of the system and the threats that impact them; and, (2) for the implementation phase to identify vulnerabilities on the services or devices. In [50], the authors extend their tool with conceptual models for expressing an IoT system during the design and implementation phases, and a class-based notation of the modeling language.
	ThingML is developed as a domain-specific modeling language which includes concepts to describe both software components and communication protocols. The formalism used is a combination of architecture models, state machines and an imperative action language [51] to model hardware and software components, and communication protocols of IoT systems. It does not model social or security components of IoT as do ASTo and ASSIST.
	IoTDSL is a Domain-Specific Language relying on a high-level rule-based language [52] for describing structural configurations and event-based semantics of devices. Event orchestration translates high-level rules into a Complex Event Processing module that evaluates and triggers runtime events, and allows simulation of user-defined configurations.
	[53] presents a virtual prototyping approach to specify and analyze IoT systems consisting of 8 Domain Specific Languages (DSLs) covering the application domain, the system and its validation that can be used to automatically detect common configuration errors and erroneous behavior. The authors apply the approach to an intelligent lighting system.
Metadata extraction	In [54], the authors extract metadata from diagrams and models of software development processes (e.g., Unified Modeling Language) to automate threat modeling, security analysis and penetration testing.
Simulators	ASSIST is an agent-based simulator of Social Internet of Things (SIoTs) [55]. Here smart objects connect with each other to form social networks. It uses an agent-based approach, defining three types: Device Agents, Human Agents and Task Agents.
	SenseSim is an agent-based and discrete event simulator for IoT [56]. It can simulate heterogeneous sensor networks to observe changes. It improves the perception of sensor networks but does not integrate security analysis.

restrained IoT networks. Several existing tools are adapted to IoT. A notable example is the MMT (Montimage Monitoring Tools) framework that can monitor and analyze the security properties of many different network environments, including fixed networks, fourth- and fifth-generation mobile networks and IoT networks. For instance:

- The authors of [57] extended MMT for analyzing IoT protocols. It also introduced an interesting approach based on supervised machine learning to preprocess and analyze the data input. These techniques can leverage the data processing speed to assure quick detection even in large scale systems with high traffic. The extended MMT framework is validated in several case studies including traditional TCP/IP (v4) network monitoring (Local Area Network, Wide Area Network, Internet monitoring), IoT using 802.15.4 and 6LoWPAN (IPv6 over Low-power Wireless Personal Area Networks) technology. In each case study, it is described how the data traces are collected, extracting the relevant attributes, handling the received data and analyzing it with respect to security.
  In particular, regarding the application to 6LoWPAN traffic, the lack of existing specific security monitoring tools pushed the authors of [58] to adapt the MMT framework to work in 6LoWPAN-based Wireless Sensor Networks (WSNs). They did this by adding several new plug-ins. A number of algorithms and techniques to detect anomalies in such networks were also applied based on supervised learning including statistical learning and information theory. Several experiments were performed that evaluated the proposed solution's applicability, extensibility and performance.
- The survey presented in [59] gives a wider review of the state of the art with respect to the IoT, taking as a platform the WSN whose sensors work with the IEEE 802.15.4 standard [60]. presents the analysis of well-known threats related to the M2M communication and the possible mitigation inside of the Wireless Sensors Networks (802.15.4/6LoWPAN), taking into account the restriction related to the resources of the available devices. Of particular interest is the analysis of the Datagram Transport Layer Security protocol and the proposed monitoring rules to validate the mitigation that has been taken. The authors found that research on IoT and Wireless Sensors Networks has been mainly focused on issues related to the standardization of the communication protocols, performance improvement and optimization of resource consumption. Research on security has been relegated, because of the low resources available on the sensors. Nevertheless, the data collected in many scenarios can be highly sensitive and must be stored and transmitted in a secure way from the origin to the destiny, in a similar way than in traditional networks. Thus, in [61], the same authors propose a solution based on the MMT framework, adding a number of techniques for

detecting misconfigurations in the communication of the sensors, and performing a series of experiments to validate the proposed monitoring solution over an IoT environment. Finally, in [62], the authors use MMT for the analysis of the 6LoWPAN traffic in the upper layer and detecting security threats over a real WSN as test bed with sensors using Datagram Transport Layer Security for protecting the communication. The contribution of this work is the development of the security rules for monitoring the communication between sensors. The security rules are based on the mitigations identified by the European Telecommunications Standards Institute (ETSI).

In [63], the authors use **Software-Defined Networks** to implement a flow-based analysis engine for home IoT networks. This approach mirrors the traffic of selected flows to a dedicated module, where flow-based metrics are analyzed to detect protocols that have known vulnerabilities. The authors also show that this approach can be easily deployed in home network equipment, protecting home automation devices such as intelligent light bulbs or surveillance cameras. Following this idea, in [64] the authors propose a complete security test bed for IoT environments supporting a set of well-known tools. The authors show how this test bed can be used not only for testing functional requirements of an IoT network, but also performing security testing and monitoring. A scenario involving port scanning of IoT applications is presented. The authors provide a list of penetration tests supported by the platform, with the aim of assessing the security of the IoT network.

As mentioned before, an online security analysis of IoT networks usually involves the processing of huge amounts of data. In general, this need makes the security analysis unfeasible onsite. To cope with this problem, in [65] the authors propose a **MapReduce-based model** for IoT monitoring. In this approach, the security events that are processed are the logs generated by each IoT component and the ones available from the network components (e.g., firewalls, routers, among others). These network events are collected in a centralized machine and processed using the Hadoop MapReduce framework [66] in order to detect out-of-bound measurements. A practical application of these tools is presented in [67], where the authors propose a secure framework applied to agricultural IoT networks. In this work, the authors capture the data exchanged between the IoT controller and the (secure) network gateway in order to analyze it using discrete wavelength transforms. Using this technique, it is possible to detect any out-of-the-normal activity, which might indicate the presence of an attack in the network. Finally, the authors also integrate recovery actions able to discard any suspicious data, reauthenticate the sensors or even reconfigure the network.

13.5.2 Software-Defined Networks (SDN) and Network Function Virtualization (NFV) Technologies

An emerging approach applied to IoT networks is to take advantage of **Software-Defined Networks** (SDN) and **Network Function Virtualization** (NFV) architecture concepts to separate the control and data layers. This allows a more flexible and cost-efficient deployment of devices, but also enables better control of the behavior and checking of the status of the network in a centralized way [68, 69]. Following this approach, Flauzac et al. propose an SDN-based architecture for IoT networks [70]. In this work, the authors use the SDN technologies to implement a Network Access Control system to enable monitoring the network endpoints. This approach uses OpenFlow [71] technologies to authenticate the network devices and dynamically deploy rules for traffic forwarding based on security policies and on the given permissions of any newly registered device. In this sense, the SDN controller is aware of all the connected devices and controls the traffic a device is allowed to send and receive.

The SDN technology allows having a trusted, centralized controller that authorizes and monitors the network. However, vulnerabilities in the SDN controller might compromise the security of the whole IoT network. In this sense, Network Function Virtualization (NFV) can help to release the pressure on the SDN. By visualizing the network components, it is possible to introduce security functions at the edge of the IoT access network, and even instantiating when needed new security controls for accessing the network. A first approach of such work is presented in [72]. Here, the authors complement the SDN approach by introducing Virtual Network Functions (VNFs) in order to provide extended functionalities to the IoT network, which comprise security functions and access control. It is important to remark that the combination of SDN and NFV techniques allowed the authors to embed an Open Network Operating System (ONOS) [73] orchestrator in their approach, allowing dynamic deployment of VNFs whenever required. A similar approach has been proposed by Salman et al. [74] that use NFV to directly introduce multiple access control models, assigning permissions at different planes of the proposed network architecture.

Following this idea, the H2020 project ANASTACIA aims to further extend the security offering for IoT network with a complete autonomic SDN and NFV-enabled IoT framework [75]. This project integrates multiple IoT-tailored tools integrated in the MMT framework that include: specially adapted DPI sniffers (called MMT-IoT [76, 77]), and monitoring agents (called MMT-Probe [78] and analogue data extractors) that extract the security events directly from the IoT network. These data are fed to a monitoring module (part of the MMT monitoring framework [79]) that performs filtering and preprocessing of the data, before proceeding with events correlation-based incident detectors, with the goal of performing an integral security analysis. Based on the security verdicts, the ANASTACIA platform has the ability to react against the detected issues, applying self-healing measures in accordance with the security policies specified for the system.

The ANASTACIA project defines a security management architecture aimed to deal with the security and privacy in NFV/SDN-enabled IoT scenarios, detailing the different planes of the architecture as well as the main architectural flows. In addition, the main IoT thread/attacks and their suggested potential detection and reaction mechanisms based on NFV-SDN are being developed.

The architecture has been conceived as a **security-enabling framework** that allows an autonomic detection of the security incidents and computation of the countermeasures. To enable these features, the architecture comprises a monitoring module that will actively observe the network and ensure its security. Figure 13.4 shows the design of the monitoring component that is composed of four principal components:

1. Data Filtering and Preprocessing Broker is an intermediate layer between the incident detector and the network agents. It is intended to perform an initial filtering and reformatting of the information captured by the network agents and feed it in a normalized format to the incident detector.
2. Incident Detector is the core component of the monitoring module. This unit analyzes the processed data from the network agents and executes the security analysis, searching for security issues and attacks.
3. Attack Signatures correspond to a database containing the set of attacks that are being monitored in the network. Despite this component is shown as a module from the incident detector, it is usually embedded in the latter.
4. Data Analysis is an AI-based module that applies machine-learning techniques on the extracted data to detect behavior anomalies.

These components rely on the data extracted by devices acting as monitoring agents of the architecture, which can be seen in the IoT network represented by the cloud in Fig. 13.4. Considering their position in the whole architecture, the monitoring agents take the role of directly interacting with the monitored network, continuously extracting information from the data plane that will be used by the components mentioned above to perform the security analysis.

Monitoring IoT-based networks introduce a particular constraint on the monitoring agents. They need to restrict the consumption of energy and use as little of the available computation capabilities as possible. The energy and computation is reduced by relegating any complex task, such as the analysis of the data and applying machine learning algorithms, to devices with more capacity.

The architecture has been tested in two different contexts involving: MEC (Mobile Edge Computing) and IoT Critical Infrastructures in Building Management Systems. In these scenarios, Distributed Denial of Service (DDoS) and IoT malware attacks were respectively tested, detailing the autonomic reaction processes to mitigate them. The performance evaluation demonstrated the feasibility of the solution to automatically monitor, detect, react and mitigate IoT cyber-attacks, enforcing proper security policies with reasonable time delays depending on type of the attack and reaction mechanisms.

Despite that the presented techniques seem to widely cover the monitoring needs of the IoT networks, new connected devices and protocols open the possibility for new attacks and introduce new security requirements. In the following are described different research efforts to address security concerns:

- The authors of [80] propose a system hardening and security monitoring solution for IoT devices to mitigate IoT security vulnerabilities and threats. The primary function is to continuously monitor the system hardening status of IoT devices. The security monitoring proposed continuously analyzes the logs generated from the logging function activated within the IoT devices to detect any anomaly. The authors give as an example of an attack the persistent SSH access requests from unauthorized external devices. This attack is detected by analyzing the logs and various response strategies are made possible such as notifying the IoT device manager or blocking the corresponding IP address.
- Similarly, [81] proposes log analysis but this time based on a semantic caching framework that uses FPGA acceleration hardware for fast processing that needs to be configured for a given data store and execution environment.
- To deal with the scalability large IoT networks, [82] proposes a solution to analyze very large amounts of data in real time and with minimal computational costs adapted for IoT networks. This is similar to a **Security Information and Event Management** (SIEM) system that monitors application and system events from different sources. In the case of IoT networks, the monitoring consists of collecting data about security events from remote devices, information sensors and network elements and their preliminary processing which includes data normalization, data filtering, data aggregation and data correlation. The

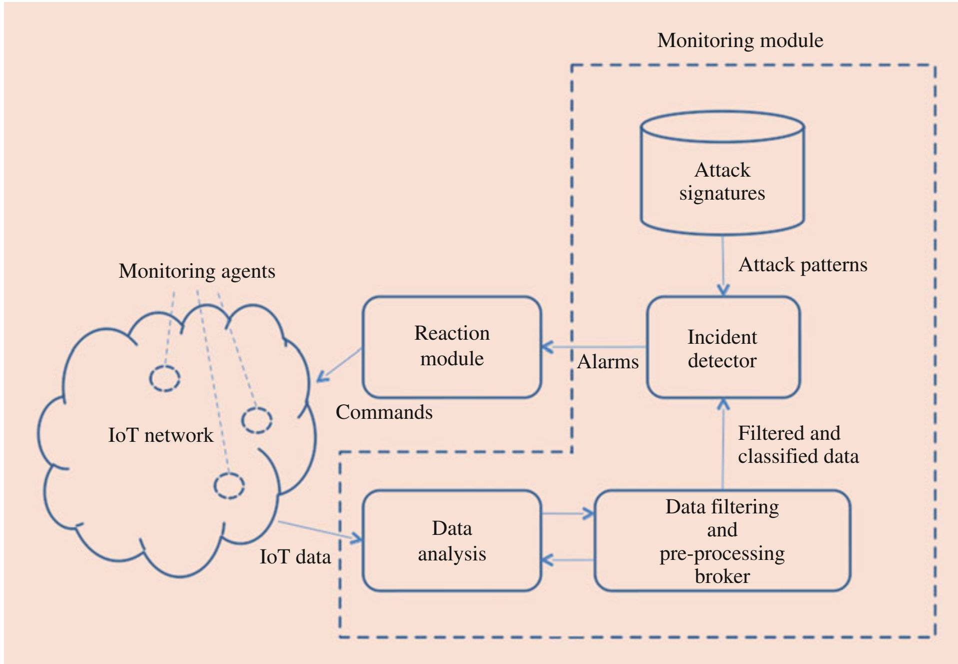

Fig. 13.4 High-level architecture of the monitoring module

results of preliminary processing are visualized so that the operators can make decisions related to the security of the IoT network; and the system uses and compares the performance of Hadoop and Spark parallel processing platforms to perform the data collection, storage, aggregation, normalization, analysis and visualization. In another paper, [83] presents a framework that integrates the Big Data processing with machine learning algorithms, analyzes a reference data set containing mobile IoT traffic data and assesses the results obtained.

- Another way of dealing with scalability is proposed in [84]. The solution consists of a decentralized multiagent system as a way to place decentralized intelligence in distributed computing, specifically by supporting computation at the level of social or business meanings. For the authors, Internet of Things (IoT) has become a major thrust in distributed computing and introduces major challenges for distributed intelligence that include: heterogeneity of IoT components; asynchronous and delay-tolerant communications and decoupled enactment; and, multiple stakeholders with subtle requirements for governance, incorporating resource usage, cooperation and privacy. Thus, IoT security solutions need to support multiple stakeholders engaging in complex interactions sometimes over highly constrained resources; but, new ways to support flexible reasoning, enactment and governance are needed that consider the social implications. Merely patching existing approaches is not enough and placing decentralized intelligence constructs such as norms at the heart of IoT-based distributed systems is required.

Two aspects that the authors identify are:

- Distribution of resources. Distribution is nominally demonstrated by diverse application areas but mainly as a convenience. In practice, distribution has been reduced in system architectures. Instead of true distributed computing, it has been economical to develop semicentralized architectures such as cloud computing.
- Different stakeholders. IoT is conducive to independent ownership and independent operation of resources. This is because IoT devices are physically distributed and cross jurisdictional boundaries and are therefore well aligned with business models in which some of the ownership is likewise spread over the stakeholders. Increasing recognition of privacy risks with the IoT brings up the need for incorporating governance within an Iota, which is possible only if one develops computational representations of the social sphere in which an Iota exists.
- A novel approach for monitoring and enforcing network policies is described in [85]. The goal is to take advantage of techniques, such as network discovery and device behavior fingerprinting, to define per-device/user network policies and enforcing them at the network edge before unwanted traffic enters or leaves the monitored network perimeter. The architecture proposed can be used for both distributing and enforcing **security policies** designed to protect simple IoT devices as well servers and workstations. It allows creating simple security applications, small enough in terms of computing resources, yet able to

increase network protection of IoT/home networks by restricting Internet access based on the device type, and able to detect and insulate network threats caused by malware or compromised devices running within the internal network. The novel contribution of this work is the idea that one can use dynamic network discovery not just to better map network devices by labeling them with a type/category, but applying to each device a comprehensive network profile based on its type. It allows moving from coarse-grained security policies, implemented by a central firewall, toward edge-based fine-grained policy enforcement tailor-made for each device type/category [86].

IoT **security monitoring** is different from IT security in many ways since it introduces new requirements. Besides the need to consider new IoT components, standards and protocols, solutions must cope with new issues. First, they need to guarantee the protection of vulnerable IoT devices with computing and energy constraints. They also need to cope with scalability issues, since securing IoT networks often involves dealing with highly distributed and numerous IoT devices. Furthermore, unforeseen interoperability and adaptability problems can appear [87] and the dynamicity of these systems requires continuously adapting to changes in the configuration and topology. Nevertheless, as pointed out by IBM, IoT systems *cannot depend on the constant integrity of every connected device to ensure the ongoing integrity of the whole system,* but need to *assume that individual devices might be compromised and still be able to function securely with one or more compromised devices* [88]. The monitoring needs to be leveraged with risk analysis to make the security solution both more efficient and effective.

Another issue, that is not considered here, is the need to monitor the **physical security** of the sensors since they are not necessarily installed in controlled environments. Here, only the security of the communications of the devices is considered. For this, in some cases, it helps that the IoT monitoring traffic patterns do not vary much, making it easier to detect anomalies. For security monitoring, it is necessary to: keep knowledge about the system up to date, such as identifying network elements (discovery); identify the role of these elements; assign a profile or a set of security policies to each type; detect breaches of policies; aggregate specialized metrics, business activity analysis, log analysis, **Deep Packet Inspection** techniques and real-time telemetry monitoring; and, enable reactions (e.g., enforcing, mitigating, notifying).

Threats on IoT devices can pose significant risks that manufacturers have not sufficiently considered. Manufacturers have been primarily concerned with rolling out new sensor devices and applications, and have not incorporated any security-by-design features. A notable example is the use of common factory default usernames and passwords that make a very large amount of deployed devices very easy to hack. This results in, for instance, the exploitation of IoT devices to perform **Mirai botnet-type attacks** [89].

The lack of automated software updates, vendor support as well as user's misconfigurations make the IoT prone to cyber-attacks. In this context, there is a need of advanced and adaptive mechanisms able to dynamically ensure the proper security levels in the IoT systems and provide system resiliency through self-healing and self-repair capabilities, thereby countering cyber-attacks and mitigating cyber-threats whenever they occur in the managed IoT network. In this sense, contextual and monitoring information obtained from the surrounded IoT environments can be used as baseline for data analysis and detection of anomalous behaviors, and in turn, infer smart control and management decisions through different actuators, agents and controllers deployed either at the edge or in the core of the IoT network. This contextual and real-time monitoring can used to deal with diverse kind of cyber-threats and IoT attacks, thereby countering them by adapting security policies and enforced configurations of the managed IoT system according to the context [90].

13.5.3 Time Series Analysis

In the case that IoT networks are used to perform periodic measurement, it is also possible to analyze the values of these measurements to detect anomalies that could be due to tampering of the sensors or the communications, i.e., through **time series analysis**. Nevertheless, it must be noted that it could be difficult to determine the root cause of a detected anomaly since it could be due to tampering or to anomalies due to physical events. To be able to determine the causes it is necessary to understand and carefully consider the characteristics of the application domain. Typical domains are health care or industrial surveillance systems.

An example of this type of analysis can be found in [91] that analyze the measures obtained from a wastewater plant. The authors apply different algorithms to detect abnormal variations in the values of the measurements over time. The method used can be based on:

- **Statistical analysis** (e.g., using the low-high pass filter method) that rely on past measurements to approximate a model of the expected behavior of the measures;
- **Probabilistic analysis** (e.g., using Hidden Markov Models, Bayesian Networks) that could be parametric or nonparametric depending if the measurements follow a certain distribution model or not;

- **Proximity-based analysis** (e.g., using the Local Outlier Factor algorithm) that rely on the distance between data measurements;
- **Clustering-based analysis** (e.g., using Hierarchical, K-means, Density-Based Spatial Clustering of Applications with Noise clustering algorithms) where measurements are separated into clusters;
- **Prediction-based analysis** (e.g., using machine learning algorithms, Deep Neural Networks, Long Short-Term Memory) that rely on past history to train a model that can predict the future values with a certain level of confidence.

The authors of [91] indicate that it is difficult to select the best algorithm since, for instance, some are better for detecting single outliers while others for detecting anomalous trends. The optimal solution, as stated by the authors" would be selecting "*a few of the proposed solutions to form a model based on an ensemble of experts. The experts' outputs would then be combined using either a majority vote approach, or a weight-based strategy, to decide which acquisition is to be classified as anomalous.*"

13.6 Tools

From the industrial point of view, the needs for solutions are compelling. A Great Bay survey carried out in 2016 [92] found that 71% of IoT Enterprises Security Professionals were not monitoring IoT devices in real time.

13.6.1 Monitoring Tools

Several **monitoring tools** or solutions specifically addressing IoT networks have been proposed in the literature but these are mainly academic, some of which are part of the work described in the previous section. Examples of such monitoring tools that exist today are Foren6 [93] and SVELTE [94]. Foren6 mainly focuses on visualizing the network topology and analyzing routing concerns. SVELTE takes a more active role, meaning that it creates additional traffic to realize their goals which might hamper resource-constrained networks.

In [95], Gartner has analyzed different vendors and identified some that propose different kinds of monitoring solutions enabling real-time visibility and control, tracking, discovery, threat detection and response in IoT networks. Gartner points out that cloud-based security services will play an indispensable role in providing IoT security due to the scale of services required: *IoT will not be viable in the long term without the cloud*. Furthermore, according to Gartner, the diversity of IoT devices and their life cycles drive hybrid security solutions for legacy and modern IoT deployments, depending on the vertical industry.

Currently, there are few vendors that offer real-time visibility and control of every network-connected IoT device. These security products are able to sniff and scan IoT networks and every connected IoT device regardless of wired/wireless technology and radio frequencies used, and independently from their location. These features are intended for providing improved IoT security assessment and awareness. They allow monitoring, tracking, alerting, detecting and responding to IoT specific threats. The vendors identified by Gartner are given in Table 13.4.

13.6.2 IoT Ecosystems

IoT service providers agree that bringing security to the IoT network is a challenging task. They try to address this challenge by integrating security analysis into the **IoT ecosystem** they offer as a product. The idea behind this is to use an already-developed and integrated IoT firmware (part of the IoT ecosystem) that is capable of sending periodic reports to a

Table 13.4 Commercial IoT monitoring tools

Vendor	Product	Web link	Basic functionality
Bastille	Enterprise Internet of Things Security	https://www.bastille.net/	Identification of threats that uses Bayesian statistics to identify anomalies, and implementation of responses
Forescout	CounterACT	https://www.forescout.com/products/counteract/	IoT visibility
Great Bay Software	Beacon	http://www.greatbaysoftware.com/products/beaconendpoint-profiler/	IoT discovery and visibility. IoT behavior monitoring
Qadium	Expander	https://qadium.com/	Visibility in IoT networks
ZingBox	IoT Guardian	http://www.zingbox.com/why-zingbox	IoT discovery, visibility and insights
Pwnie Express	Pulse IoT Security Platform	https://www.pwnieexpress.com/products/pulse	Discover and track monitor devices. Device threat detection that performs device discovery to detect rogue devices, vulnerability scans and policy-infringing connections.

behavior analysis engine (also part of the ecosystem). This approach facilitates the securing process of the IoT network by passively analyzing the behavior of all the involved devices, raising alerts in case an anomaly is detected. In this market, one can find two principal IoT-monitoring solutions integrated within their IoT services: Microsoft Azure Sphere [96] and Amazon AWS IoT Services [97].

Microsoft Azure Sphere is a complete IoT ecosystem built to bring security to the IoT devices. It provides certified microcontrollers that integrate security layers and security events collection to Azure Cloud services. The pieces of hardware are complemented with Azure Sphere OS, an IoT firmware designed to offer multiple layers of security based on Window and Linux kernels. All these technologies are completed with cloud services by offering Azure Sphere Security Service. This last service acts as a centralized trust and security service providing communication privacy, device authentication, failure reports, threats responses and centralized device updates.

Likewise, Amazon offers a variety of security tools for IoT networks. They provide FreeRTOS as an IoT operating system that integrates a set of security functionalities (for communications) and additional libraries to connect the device to the Amazon AWS cloud service. Since FreeRTOS has been conceived as an open source IoT firmware, the usage of AWS libraries is optional, but they enable the usage of the cloud-enhanced tools AWS IoT Device Management (AIDM) and AWS IoT Device Defender (AIDD). AIDM helps executing maintenance activities on the devices (such as upgrading software, defining access policies); while AIDD expands the security options by bringing audit capabilities to the IoT configurations, behavioral analysis of the devices and alert services (connected with Amazon AWS CloudWatch) for informing on the detection of abnormal behavior.

Stemming from the before-mentioned ANASTACIA project, MMT-IoT [77] was developed to fill the identified missing gaps that would allow obtaining a more efficient security monitoring solution for resource constrained IoT networks. In this context, MMT-IoT has been developed to target IoT technology and allow capturing IoT network traffic near the IoT devices and analyze this traffic to detect potential attacks. This solution is being industrialized and will be commercialized by the end of 2019.

13.7 Concrete Example: IoT Security Monitoring and Test on Fed4Fire + Platforms

The work presented in [76] provides a concrete example of monitoring the security of an IoT platform. Experiments were conducted using the MMT-IoT security analysis solution running on a **Fed4Fire-Plus IoT platform** provided by IMEC of Belgium and named Virtual Wall – w.iLab. The results obtained allowed evaluating the capability of the techniques used, namely Deep Packet and Session Inspection of the IoT protocol exchanges, behavior analysis and rule-based analysis using the formal specification of temporal logic.

MMT is a monitoring framework developed by Montimage, and MMT-IoT is a tool based on this framework to monitor and analyze the security and performance of IoT networks. It is a security tool designed to bring awareness on the dynamic behavior of the IoT system and devices and assure that the security requirements of the IoT network and applications are respected in industrial environments. MMT-IoT captures IoT radio network traffic near the IoT devices and analyzes it to detect potential attacks, anomalies and misbehaviors. In this work, the Fed4Fire industrial test bed made it possible to deploy MMT-IoT on real-life operational scenarios to validate the security detection capabilities of given properties and of deviations from normal expected behavior, as well as the execution of initial scalability tests.

The results obtained effectively demonstrate the feasibility and validate the two main contributions. First, they allowed determining the necessary adaptations to deploy MMT-IoT on an industrial IoT platform and run the tool on the IoT devices if this platform. Second, the software deployment allowed carrying out preliminary tests of the platform and performing initial validation and scalability testing on a real environment. To this end, the authors designed and implemented three security and scalability test scenarios with one or more clients. These results are being used to prepare a new experimental phase also involving another Fed4Fire + platform proposed by IJS of Slovakia and called LOG-a-TEC.

13.7.1 Montimage Monitoring Tool (MMT) Designed for Monitoring IoT Networks

The Montimage Monitoring Tool (MMT) [35] is a modular monitoring framework that can detect behavior, security and performance incidents based on a set of formal properties (written in XML) and built-in functions (written in C or any script or interpreted language). The formal properties can specify known vulnerabilities and attacks, or expected **behavior** whose deviation from it could be due to a vulnerability, an anomaly, a malfunction, or an attack; and the built-in functions that allow more sophisticated analysis based, for instance, on statistics, correlation with cyber-threat intelligence, and artificial intelligence and machine learning techniques. MMT enables real-time data capture, metadata extraction, correlation of data from different sources (network traffic, application and operating system traces and logs), and it performs complex event processing and distributed analysis. It uses time-based logic to detect given (expected or abnormal) security properties and a statistical analysis based on trends analysis or machine learning to detect previously unknown malicious activities and behaviors. It is relatively easy to extend by adding new: integrated properties and functions; plug-ins to analyze any protocol or structured message; new dashboards for the visualization of the data, statistics and alarms; and, instructions for triggering reactions (e.g., mitigating or blocking attacks).

In order to properly adapt this approach (initially designed for traditional Ethernet networks) to IoT networks, it was necessary to divide the network extractor (sniffer) into two parts: the MMT-**IoT Sniffer** (a Contiki-based IoT device) and the MMT-**IoT Bridge** (a Linux-based tool). The first is the IoT endpoint which is responsible for collecting the communication packets and transferring them via an USB line to a more powerful machine. The latter retrieves the packets from the USB line and injects them (encapsulated using the ZEP protocol) into the loopback interface of the machine, thus making the packets ready to be analyzed by the MMT-Probe and MMT-Security tools of the framework. Figure 13.5 summarizes the general architecture of the solution.

Concerning the MMT-IoT Sniffer, the implementation of this architecture was achieved by introducing modifications in the network drivers to make the sniffing feature work. Such modifications involved three main areas:

- Radio driver in promiscuous mode: This modification was done to avoid the dropping of packets by the Contiki kernel.
- Avoiding dropping packets with bad checksum: By default, the radio driver reads the packets and performs a Cyclic Redundancy Check (CRC) to detect potential transmission failures. If this check fails, the packer is discarded to avoid processing an incorrectly formatted packet and save energy. This behavior was changed, since the sniffing solution needs to extract all the packets on the medium whether they are correct or not.
- Inserting callbacks to redirect the received packet: A sniffer is a passive network element; therefore, once the packet is received on the radio driver layer, it is transferred via callbacks directly to the application layer. This behavior bypasses the Contiki network processing and redirects the packets immediately using the USB line, saving energy in the sniffer device. The structure of the inserted callbacks is depicted in Fig. 13.6.

Finally, the MMT-IoT Bridge is responsible for capturing the packets sent through the USB line and making them available for the security analysis performed by the MMT-Probe and MMT-Security; both modules that form part of the MMT framework.

This security analysis is performed by a set of security rules, previously defined by a network security expert, which codify the set of network events and extracted metadata that need to be correlated for detecting security issues. It is important to note that computation complexity of detecting an attack is given by the rule itself: complex attacks require more complex rules which correlate a higher number of network events. Considering this, the computation complexity will be managed by the MMT-Probe, and not the

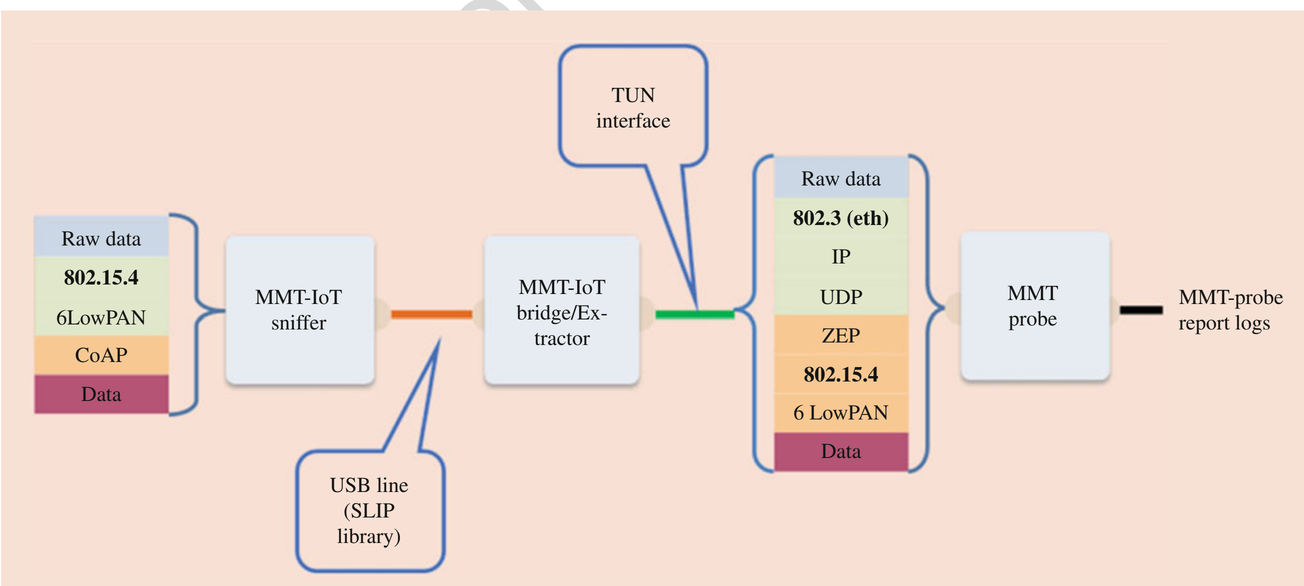

Fig. 13.5 General architecture of the MMT-IoT solution, MMT-Probe and MMT-Security

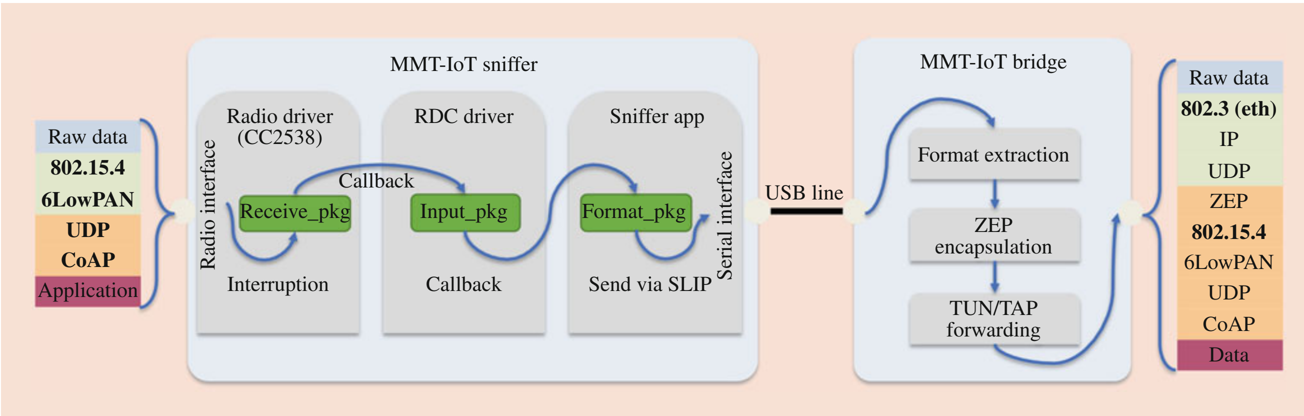

Fig. 13.6 Internal details of the MMT-IoT solution

MMT-IoT module, whose role is only to redirect the traffic to the MMT-Probe. This is why neither the MMT-IoT Sniffer nor the MMT-IoT Bridge components contain any complex processing logic that is dealt with by the security analysis performed by MMT-Probe.

13.7.2 Description of the Fed4Fire + Test Beds

Future Internet Research and Experimentation (FIRE) was launched by the European Horizon 2020 research program to enable to carry out research activity and experiments. Experiments are considered to be a key factor for the continued impact and growth of the European Internet industry. Each project in the Future Internet Research and Experimentation (FIRE) initiative targets a specific community within the Future Internet ecosystem. Through the federation of these infrastructures, innovative experiments become possible that break the boundaries of these domains. Besides, infrastructure developers can utilize common tools of the federation, allowing them to focus more on their core test bed activities.

In this sense, Fed4FIRE+ is a project under the European Union Program Horizon 2020, offering the largest worldwide federation of Next-Generation Internet (NGI) test beds. These provide open and reliable facilities supporting a wide variety of different research and innovation communities and initiatives in Europe, including the fifth-generation mobile networks (5G) Private–Public Partnership (PPP) projects.

In the work described here, the platforms LOG-a-TEC and Virtual Wall (w.iLab) that are part of Fed4FIRE+ were considered. It must be noted that only Virtual Wall (w.iLab) was used to perform the experiments described here. In the case of LOG-a-TEC, only a feasibility study was made and the experiments on this platform will be performed at a later stage. Following is a brief description of each platform:

- LOG-a-TEC: LOG-a-TEC is proposed by IJS, Slovenia [37]. It is composed of several different radio technologies that enable dense and heterogeneous IoT, Machine Type Communication (MTC) and fifth-generation (5G) mobile network experimentation. Specially developed embedded wireless sensor nodes can host four different wireless technologies and seven types of wireless transceivers. In order to enable different experiments in combined indoor/outdoor environments using heterogeneous wireless technologies, the test bed is deployed within JSI's premises and outside in the surrounding park and on the walls of the buildings. The feasibility of using this platform to carry out experiments has been validated and a new experimentation phase will allow performing the scenarios described and demonstrate the genericity of the monitoring solution.
- Virtual Wall: The w.iLab platform [34] is an IoT and 5G emulation test bed that allows running experiments on nodes on real IoT deployments. This platform was designed by the IMEC, Belgium. It provides *bare metal* access to its nodes, in other words, it gives root access to physical machines that will be used to run the experiment. This allows the experimenter to have full control of the nodes on the test bed. The deployment of the MMT-IoT and MMT-Probe software and the execution of the tests are performed remotely without requiring major interventions from the operators. For this, credentials were created on the iMinds platform and performed a reservation of the Intel NUC nodes from the Datacenter or of the platform. The jFED-Experimenter tool was required to design an experiment to access these nodes.

13.7.3 Experimental Evaluation

Considering these test beds, the authors used the w.iLab platform to deploy the MMT-IoT Sniffer and the MMT-Probe solutions. In this way, they were able to use

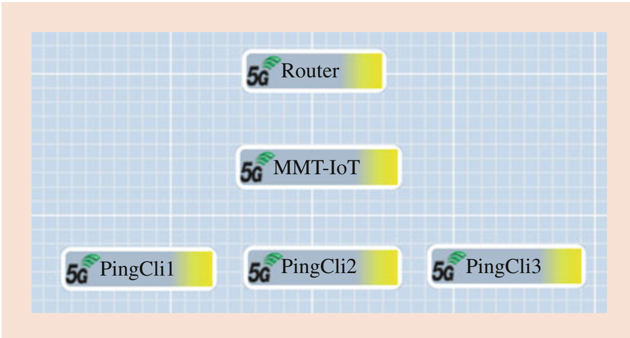

Fig. 13.7 Deployment of the MMT-IoT solution in the w.iLab platform

the w.iLab t.1 platform to evaluate the scalability of these by overloading them. By performing the extraction of the packets from an IoT network, this experimentation pursued two principal subobjectives: (1) perform an initial Deep **Packet Inspection**-based security analysis of the IoT network traffic; and (2) determine the maximum throughput a single instance of MMT-IoT Sniffer can handle. To achieve these objectives, the authors deployed a set of IoT devices as shown in Fig. 13.7. In this deployment, three types of devices were used:

- Ping Client: An emulated IoT sensor programmed to attack the server. For the emulation purposes, a client that performs a "ping" to the IoT router was used. However, in real life, a client can be any device generating some type of traffic.
- **IoT Router**: A gateway running a routing protocol to allow communications within the IoT network.
- MMT-IoT: A node running the Montimage software under test.

The deployment described above was used to perform initial validations and scalability tests in scenarios that contain respectively 1, 2, 3 malicious clients. These configurations allowed performing both objectives previously mentioned:

- The security analysis validation, by means of determining the number of detected attacks;
- The scalability of the MMT-IoT solution, by means of analyzing the number of extracted packets in each scenario. This aims to determine the amount of information an IoT sniffer is capable of handling at a time.

To deploy the testing scenarios, the nodes provided by the w.iLab Platform were used, each one composed of a Linux machine with two Zolertia Re-Mote IoT nodes. On each node, the Zolertia Remote nodes were used to install the corresponding device type (in form of an IoT firmware) and generate the test traffic. Additionally, the MMT-IoT Bridge, MMT-Probe and MMT-Security software were installed on the MMT-IoT Linux machine. This was done in order to read the packets extracted by the IoT sniffer and perform the security analysis on the same node.

The Ping Client IoT sensors were configured to trigger the attack every 10 seconds. At each triggering, the client sent a burst of 10 ICMP ping packets equally spaced within a second. Additionally, an RPL router image was deployed in the IoT-Router machine in order to allow packets to flow through the network.

All the MMT software was deployed in the MMT-IoT machine, including the MMT-IoT sniffer (in the Zolertia Remote connected to that node), the MMT-IoT Bridge (running on the same NUC machine) and the MMT-Probe (also running on the NUC machine). This latter was the component in charge of analyzing the extracted packets and performing attack detection according to a rule previously defined: One should not allow more than 2 ICMP ping packets per second on an IoT network. This value used in the rule considers that, in for instance IPv6 and 6LowPAN networks, ICMP traffic is needed to keep the network alive (e.g., ping packets). In this sense, the rule allows a fair amount of ICMP packets to run through the network without raising an attack alert. This is done to reduce the number of false positives detected by MMT. Using this rule, MMT-Probe was capable of detecting the occurrence of three or more ICMP packets as an attack, generating a report in the MMT-Probe's logs. Besides detecting anomalous quantity of packets, the rule-based technique can also be used to detect anomalies in the type of ICMP packets that are being exchanged.

Each scenario was executed continuously during 5 min, in order to generate enough traffic for later analysis. The packets extracted with MMT-IoT Sniffer (using the tcpdump tool) and the MMT-Probe logs are used to check the number of detected attacks in the scenario.

13.7.4 Results Obtained

Figures 13.8, 13.9 and 13.10 show the results of the execution of the three scenarios, respectively with 1, 2 and 3 clients. In these figures, one can observe peaks each 10 seconds. These peaks correspond to the automatic triggering of the attacks, in other words, they show the moment when the clients started to send the ICMP ping packets. In these particular instances, a raise in the extracted traffic was observed since there was more data available to be processed. In the three-client scenario, after 3 min of execution one can see that the peaks appear more often. The authors conjecture that this behavior is due to some type of de-synchronization between the three clients, and the different attacks appear more frequently.

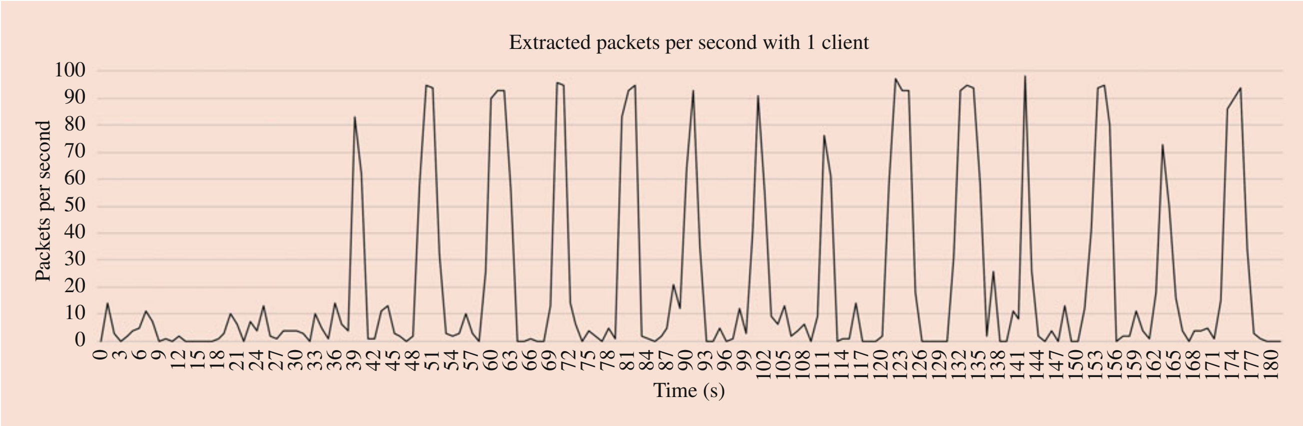

Fig. 13.8 Throughput extracted using MMT-IoT and 1 client

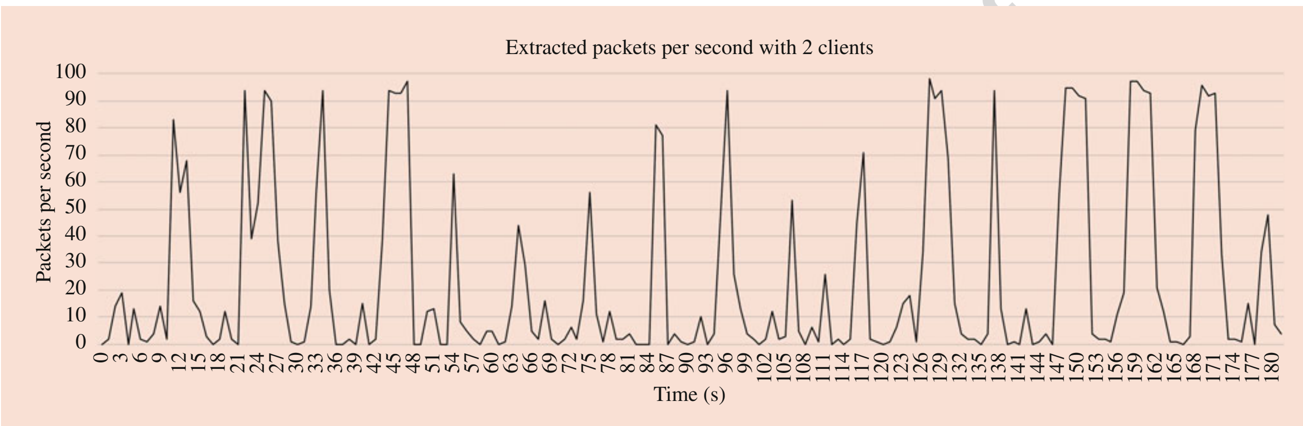

Fig. 13.9 Throughput extracted using MMT-IoT and 2 clients

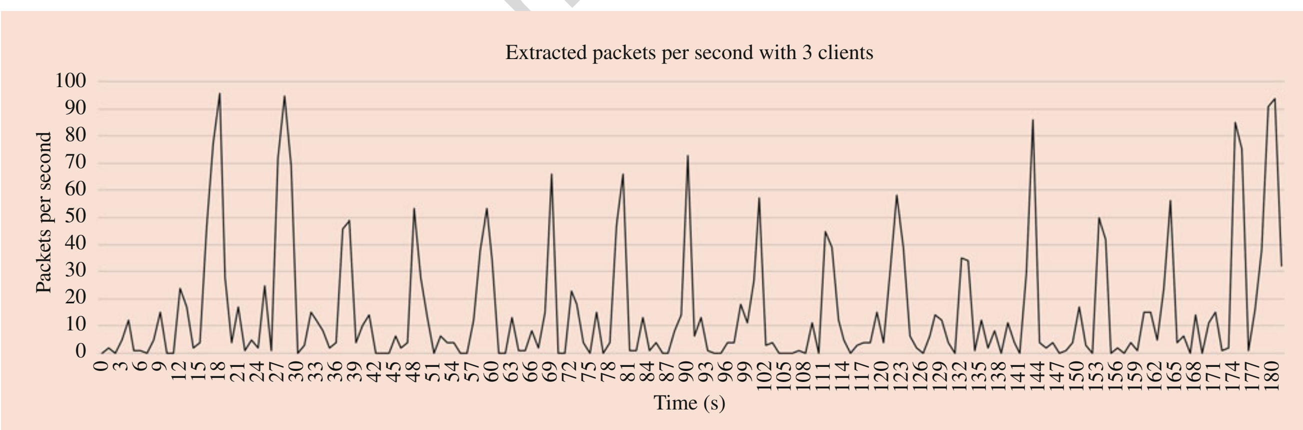

Fig. 13.10 Throughput extracted using MMT-IoT and 3 clients

An interesting observation is the limit of the extracted packets per second. Despite the fact that in the scenario more and more clients are added, and thus more traffic is generated, the maximum number of packets extracted remained practically the same: around 95 packets per second. This opens the possibility of performing experiments to answer the following questions:

- How does the packet size impact the number of packets extracted by the MMT-IoT?

- Given the MTU of the IoT network, what is the upper limit of the throughput extracted by MMT-IoT?

Finally, by analyzing the logs of the MMT-Probe, it was possible to count the number of attack detected. In the scenario with 1 attacking client, MMT-Probe detected 183 attacks; with 2 clients, MMT-Probe detected 1046 attacks; and with 3 clients, MMT-Probe detected 968 attacks. These numbers allowed validating the applicability of the MMT solution in IoT network environments.

In the case of a single attacker, MMT-Probe was capable of analyzing the packets extracted by the MMT-IoT Sniffer and detects a simple security threat inside an IoT network.

The work presented here allowed better understanding the concrete use of the monitoring techniques implemented by the MMT-IoT tool, namely the capture of the wireless protocol communications. It also allowed testing the techniques provided by the MMT framework, namely Deep Packet and Flow Inspection, Complex Event Processing, temporal logic-based rule detection, trend-based statistical analysis, machine learning algorithms, etc. It further allowed understanding how the deployment on one of the Fed4Fire + test bed platforms can be done, and the feasibility of performing the tests on another test bed platform. These tests allowed validating a proof-of-concept version of MMT-IoT on a real IoT environment. The results allow identifying potential optimizations in the techniques used and improve the detection algorithms, aiming to increase the effectiveness of the techniques.

13.8 Conclusion

Two of the biggest challenges related to IoT networks concern security and privacy. The requirements and techniques related to these issues are well understood by the research community and the different stakeholders but this does not necessarily translate into commercially secure IoT products. For this, regulations need to be stricter and their enforcement needs to be guaranteed. Having said this, the more technical challenges are:

13.8.1 Concerning Privacy

There is no comprehensive methodology or framework that ensures privacy in an IoT environment for a large class of applications and heterogeneous devices. Adapting network virtualization and, in particular, Software-Defined Networking (SDN) with its centralized nature, can help introduce security and privacy functions. Nevertheless, these techniques would need to, in many use cases, to deal with huge amounts of data that would forcibly impact the latency and performance. Furthermore, cryptography is being adapted to IoT by introducing new lightweight encryption to secure the IoT communications and lightweight security protocols.

13.8.2 Concerning Energy Consumption, Processing Capability and Storage Space

Optimizing the use of energy, processing and storage is a constant requirement that is even more challenging when security and privacy functions are introduced. Distribution and parallelization of computations, optimized using, for instance, Named Function Networking (NFN) paradigms (e.g., [98, 99]) or micro services (e.g., [100]) that would allow distributing the computations but at the same time reduce redundancy in the computations.

13.8.3 Concerning Routing

Secure routing and forwarding needs to consider IoT requirements. P4 (e.g., [101]) that allows controlling the data plane traffic of a packet forwarding device could be adapted to IoT devices. New security protocols or modified existing ones also need to consider the specific requirements of IoT. Currently, Wireless Sensor Networks use many protocols that are not secure.

Furthermore, the infrastructure-less characteristic and other requirements, such as difficult-to-access devices in the field, introduce the possibility of intrusions that need to be detected and mitigated.

13.8.4 Concerning Intrusion Detection and Prevention

Existing intrusion detection and prevention systems are designed essentially for analyzing the Internet protocols, but there is the need for detecting and acting on the IoT network radio part itself. Attacks (in other words, insider attacks since there is no real boundary) that directly access IoT devices can only be detected if the signals are monitored and analyzed directly on the IoT network (as done in [76, 77] presented before) and not after the IoT/Internet gateway or bridge.

Anomaly detection can also be used to detect tampering of IoT or Wireless Sensor Networks that are used to gather time series data. For this, it is necessary to combine statistical and trend analysis of the measures with expert knowledge. Expert knowledge is needed to take into account what the measures represent, what are the expected values and any existing correlation between the different measures in the case where several different types of measures are made (as done in [91]).

Mitigation or prevention by blocking messages is not possible in IoT communication environments. Mitigation scenarios need to be considered that depend on the application domain. For instance, by introducing device redundancy to switch from compromised to uncompromised, honeypots to redirect detected malicious traffic and correlating IoT messages with the corresponding Internet traffic so that it can be blocked. Mitigation also concerns assuring that the system continues to function at all times (in other words, the system is robust and resilient) even when faults occur in the IoT network or devices due to bugs in the software or hardware, provoked by attacks, or resulting from the depletion of energy of certain devices.

The types of attacks that need to be considered are for instance, Denial of Services, insider attacks and data exfiltration. Machine learning techniques [102] can be used but they need to be adapted to IoT constraints: large networks without boundaries, limited access to devices, and limited resources (in other words, energy, CPU and memory).

References

1. Sivaraman, V., Gharakheili, H.H., Vishwanath, A., Boreli, R., Mehani, O.: Network-level security and privacy control for smart-home IoT devices. In: In 2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), pp. 163–167. IEEE (2015)
2. Burhan, M., Rehman, R.A., Khan, B., Kim, B.-S.: IoT elements, layered architectures and security issues: A comprehensive survey. Sensors. **18**(9), 2796 (2018)
3. Kozlov, D., Veijalainen, J., Ali, Y.: Security and privacy threats in IoT architectures. In: Proceedings of the 7th International Conference on Body Area Networks, Oslo, Norway, 24–26 February 2012, pp. 256–262. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), Brussels, Belgium (2012)
4. Gupta, S., Gupta, B.B.: Cross-Site Scripting (XSS) attacks and defense mechanisms: Classification and state-of-the-art. Int. J. Syst. Assur. Eng. Manag. **8**, 512–530 (2017)
5. Miao, Y., Bu, Y.X.: Research on the architecture and key technology of Internet of Things (IoT) applied on smart grid. In: Proceedings of the 2010 International Conference on Advances in Energy Engineering (ICAEE), Beijing, China, 19–20 June 2010, pp. 69–72 (2010)
6. TechTarget: Exploit Attack in Network Layer. Online: http://searchsecurity.techtarget.com/definition/exploit
7. Puthal, D., Nepal, S., Ranjan, R., Chen, J.: Threats to networking cloud and edge datacenters in the Internet of Things. IEEE Cloud Comput. **3**, 64–71 (2016)
8. Said, O., Masud, M.: Towards Internet of things: Survey and future vision. Int. J. Comput. Netw. **5**, 1–17 (2013)
9. Xiaohui, X.: Study on security problems and key technologies of the Internet of things. In: Proceedings of the 5th International Conference on Computational and Information Sciences (ICCIS), Shiyan, China, 21–23 June 2013, pp. 407–410 (2013)
10. Brumley, D., Boneh, D.: Remote timing attacks are practical. Comput. Netw. **48**, 701–716 (2005)
11. Prabhakar, S.: Network security in digitalization: Attacks and defence. Int. J. Res. Comput. Appl. Robot. **5**, 46–52 (2017)
12. Cisco white paper: The Internet of Things Reference Model. 2014; Online: http://cdn.iotwf.com/resources/71/IoT_Reference_Model_White_Paper_June_4_2014.pdf
13. Weyrich, M., Ebert, C.: Reference architectures for the Internet of Things. IEEE Softw. **33**(1), 112–116 (2016)
14. Bader, S.R., Maleshkova, M., Lohmann, S.: Structuring reference architectures for the industrial Internet of Things. Future Internet. **11**(7), 151 (2019)
15. Sadeghi, A.-R., Wachsmann, C., Waidner, M.: Security and privacy challenges in industrial internet of things. DAC. **54**(1-54), 6 (2015)
16. Ouchani, S.: Ensuring the Functional Correctness of IoT through Formal Modeling and Verification. MEDI2018: pp. 401–417 (2018)
17. Kwiatkowska, M., Norman, G., Parker, D.: PRISM 4.0: Verification of probabilistic real-time systems. In: Gopalakrishnan, G., Qadeer, S. (eds.) CAV 2011. LNCS, vol. 6806, pp. 585–591. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22110-1_47
18. Baier, C., Katoen, J.P.: Principles of Model Checking. The MIT Press, New York (2008)
19. Beaulaton, D., Said, N.B., Cristescu, I., Fleurquin, R., Legay, A., et al.: A language for analyzing security of IOT systems. In: SoSE 2018 – 13th Annual Conference on System of Systems Engineering, Jun 2018, pp. 37–44, Paris. https://doi.org/10.1109/SYSOSE.2018.8428704. hal-01960860
20. Attie, P.C., Bensalem, S., Bozga, M., Jaber, M., Sifakis, J., Zaraket, F.A.: Global and local deadlock freedom in BIP. ACM Trans. Softw. Eng. Methodol. **26**(3), 9:1–9:48 (2018)
21. Mohsin, M., Anwar, Z., Husari, G., Al-Shaer, E., Rahman, M.A.: IoTSAT: A formal framework for security analysis of the Internet of Things (IoT). In: 2016 IEEE Conference on Communications and Network Security (CNS), pp. 180–188 (October 2016)
22. Kammüller, F.: Formal modeling and analysis with humans in infrastructures for IoT health care systems. In: Tryfonas, T. (ed.) HAS 2017. LNCS, vol. 10292, pp. 339–352. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-58460-7_24
23. Wenzel, M., Paulson, L.C., Nipkow, T.: The Isabelle framework. TPHOLs 2008: pp. 33–38
24. Torjusen, A., Abie, H., Paintsil, E., Trcek, D., Skomedal, Å.: Towards run-time verification of adaptive security for IoT in eHealth. In: ECSA workshops 2014: 4:1–4:8 (2014)
25. Riahi, A., Challal, Y., Natalizio, E., Chtourou, Z., Bouabdallah, A.: A systemic approach for IoT Security. DCOSS, 2013, Boston. pp. 351–355 (2013). https://doi.org/10.1109/DCOSS.2013.78. hal-00868362
26. Challal, Y.: Internet of Things security: Towards a cognitive and systemic approach. HDR Thesis, Université de Technologie de Compiègne (2012)
27. Kamel, S., Hegazi, N.H.: A proposed model of IoT security management system based on a study of Internet of Things (IoT) Security. Int. J. Sci. Eng. Res. **9**, 1227–1244 (2018)
28. Kakanakov, N., Shopov, M.: Adaptive models for security and data protection in IoT with Cloud technologies. MIPRO, 1001–1004 (2017)
29. Hussein, M., Li, S., Radermacher, A.: Model-driven development of adaptive IoT systems. Models (Satellite Events), 17–23 (2017)
30. Friedenthal, S., Moore, A., Steiner, R.: A practical guide to SysML: The systems modeling language, 3rd edn. Morgan Kaufmann / The OMG Press (2016) ISBN 978-0-12-800202-5
31. Costa, B., Pires, P.F.: Flávia Coimbra Delicato: Modeling IoT applications with SysML4IoT. SEAA, 157–164 (2016)
32. Mezghani, E., Exposito, E., Drira, K.: An autonomic cognitive pattern for Smart IoT-based system manageability: Application to comorbidity management. ACM Trans. Internet Techn. **19**(1), 8: 1–8:17 (2019)

33. Mashal, I., Alsaryrah, O., Chung, T.Y., Yang, C.Z., Kuo, W.H., Agrawal, D.P.: Choices for interaction with things on Internet and underlying issues. Ad Hoc Netw. **28**, 68–90 (2015)
34. Suo, H., Wan, J., Zou, C., Liu, J.: Security in the Internet of things: A review. In: Proceedings of the 2012 International Conference on Computer Science and Electronics Engineering (ICCSEE), Hangzhou, China, 23–25 March 2012, Vol. 3, pp. 648–651 (2012)
35. Bharathi, M.V.; Tanguturi, R.C.; Jayakumar, C.; Selvamani, K.: Node capture attack in Wireless Sensor Network: A survey. In: Proceedings of the 2012 IEEE International Conference on Computational Intelligence & Computing Research (ICCIC), Coimbatore, India, 18–20 December 2012, pp. 1–3 (2012)
36. Conti, M., Dragoni, N., Lesyk, V.: A survey of man in the middle attacks. IEEE Commun. Surv. Tutor. **18**, 2027–2051 (2016)
37. Ali, B., Awad, A.I.: Cyber and physical security vulnerability assessment for IoT-based smart homes. Sensors. **18**, 817 (2018)
38. Darwish, D.: Improved layered architecture for Internet of Things. Int. J. Comput. Acad. Res. (IJCAR). **4**, 214–223 (2015)
39. Sanzgiri, A., Dasgupta, D.: Classification of insider threat detection techniques. In: Proceedings of the 11th Annual Cyber and Information Security Research Conference, Oak Ridge, TN, USA, 5–7 April 2016, p. 25. ACM, New York (2016)
40. Nurse, J.R., Erola, A., Agrafiotis, I., Goldsmith, M., Creese, S.: Smart insiders: Exploring the threat from insiders using the Internet-of-things. In: Proceedings of the 2015 International Workshop on Secure Internet of Things (SIoT), Vienna, Austria, 21–25 September 2015, pp. 5–14 (2015)
41. Madakam, S., Ramaswamy, R., Tripathi, S.: Internet of Things (IoT): A literature review. J. Comput. Commun. **3**, 164 (2015)
42. Khan, R., Khan, S.U., Zaheer, R., Khan, S.: Future Internet: The Internet of things architecture, possible applications and key challenges. In: Proceedings of the 2012 10th International Conference on Frontiers of Information Technology (FIT), Islamabad, India, 17–19 December 2012, pp. 257–260 (2012)
43. Sethi, P., Sarangi, S.R.: Internet of Things: Architectures, protocols, and applications. J. Electr. Comput. Eng. **2017**, 9324035 (2017)
44. Ashraf, Q.M., Habaebi, M.H.: Autonomic schemes for threat mitigation in Internet of Things. J. Netw. Comput. Appl. **49**, 112–127 (2015)
45. Canzanese, R., Kam, M., Mancoridis, S.: Toward an automatic, online behavioral malware classification system. In: Proceedings of the IEEE 7th International Conference on Self-Adaptive and Self-Organizing Systems (SASO), Philadelphia, 9–13 September 2013, pp. 111–120 (2013)
46. TechTarget: Business Logic Attack. Available online: http://whatis.techtarget.com/definition/business-logic-attack. Accessed 30 Aug 2019
47. Bilge, L., Dumitras, T.: Before we knew it: An empirical study of zero-day attacks in the real world. In: Proceedings of the 2012 ACM Conference on Computer and Communications Security, Raleigh, NC, USA, 16–18 October 2012, pp. 833–844. ACM, New York (2012)
48. Kaur, R., Singh, M.: A survey on zero-day polymorphic worm detection techniques. IEEE Commun. Surv. Tutor. **16**, 1520–1549 (2014)
49. Mavropoulos, O., Mouratidis, H., Fish, A., Panaousis, E.A.: ASTo: A tool for security analysis of IoT systems. SERA, 395–400 (2017)
50. Mavropoulos, O., Mouratidis, H., Fish, A., Panaousis, E.: Apparatus: A framework for security analysis in internet of things systems. Ad Hoc Netw. **92** (2019)
51. Vasilevskiy, A., Morin, B., Haugen, O., Evensen, P.: Agile development of home automation system with thingml. In: 2016 IEEE 14th International Conference on Industrial Informatics (INDIN), (2016)
52. Amrani, M., Gilson, F., Englebert, V.: Complex event processing for user-centric management of IoT systems. In: Chapter in Communications in Computer and Information Science. Springer (2018) https://doi.org/10.1007/978-3-319-94764-8_18
53. Verriet, J., Buit, L., Doornbos, R., Huijbrechts, B., Sevo, K., Sleuters, J., Verberkt, M.: Virtual Prototyping of Large-scale IoT Control Systems using Domain-specific Languages. MODELSWARD 2019, pp. 229–239
54. Ankele, R., Marksteiner, S., Nahrgang, K., Vallant, H.: Requirements and Recommendations for IoT/IIoT Models to automate Security Assurance through Threat Modeling, Security Analysis and Penetration Testing, pp. 102:1–102:8. ARES (2019)
55. Kasnesis, P., Toumanidis, L., Kogias, D., Patrikakis, C.Z., Venieris, I.S.: Assist: An agent-based siot simulator. In: 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT), vol. 2016, pp. 353–358. IEEE
56. Dyk, M., Najgebauer, A., Pierzchala, D.: Sensesim: An agent-based and discrete event simulator for wireless sensor networks and the internet of things. In: 2015 IEEE 2nd World Forum on Internet of Things (WFIoT) (2015)
57. Hoa La, V.: Security Monitoring for Network Protocols and Applications. PhD Thesis, Université de Paris-Saclay, October 2016, France. https://tel.archives-ouvertes.fr/tel-01782396/document
58. Hoa La, V., Fuentes-Samaniego, R., Cavalli, A.: A novel monitoring solution for 6LoWPAN-based Wireless Sensor Networks. In: Proceedings of the 22nd Asia-Pacific Conference on Communications (APCC 2016), August 2016, Yogyakarta, Indonesia (2016)
59. Fuentes-Samaniego, R., Cavalli, A., Nolazco-Flores, J., Baliosian, J.: A survey on wireless sensors networks security based on a layered approach. In: Proceedings of the 13th International Conference on Wired/Wireless Internet Communications (WWIC 2015), pp. 77, Chapter 1. Introduction - 93, Malaga, Spain (May 2015)
60. Fuentes-Samaniego, R., Cavalli, A., Nolazco-Flores, J.: An analysis of secure M2M communication in WSNs using DTLS. In: Proceedings of the 2nd IEEE International Workshop on Security Testing and Monitoring (STAM 2016), Nara, Japan (June 2016)
61. Fuentes-Samaniego, R. A.: Wireless Sensors Networks (WSN) monitoring – Application to secure interoperability. PhD thesis, Université Paris-Saclay, February 2017, France (2017)
62. Fuentes-Samaniego, R.A., Hoa La, V., Cavalli, A.R., Nolazco-Flores, J.A., Ramírez-Velarde, R.V.: A monitoring-based approach for WSN security using IEEE-802.15.4/6LowPAN and DTLS communication. IJAACS. **12**(3), 218–243 (2019)
63. Sivanathan, A., Sherratt, D., Gharakheili, H.H., Sivaraman, V., Vishwanath, A.: Low-cost flow-based security solutions for smart-home IoT devices. In: 2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), pp. 1–6. IEEE (2016)
64. Siboni, S., Sachidananda, V., Shabtai, A., Elovici, Y.: Security Test bed for the Internet of Things. arXiv preprint arXiv:1610.05971 (2016)
65. Saenko, I., Kotenko, I., Kushnerevich, A.: Parallel processing of big heterogeneous data for security monitoring of IoT Networks. In: 2017 25th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP), pp. 329–336. IEEE (2017)
66. White, T.: Hadoop – The Definitive Guide: MapReduce for the Cloud. O'Reilly 2009, ISBN 978-0-596-52197-4, pp. I–XIX, 1–501
67. Pacheco, J., Hariri, S.: Anomaly behavior analysis for IoT sensors. Trans. Emerg. Telecommun. Technol. **29**(4), e3188 (2018)
68. Raghavan, B., Casado, M., Koponen, T., Ratnasamy, S., Ghodsi, A., Shenker, S.: Software-defined internet architecture: Decoupling architecture from infrastructure. In: Proceedings of the 11th ACM Workshop on Hot Topics in Networks, pp. 43–48. ACM (2012)

69. Ahmad, I., Namal, S., Ylianttila, M., Gurtov, A.: Security in software defined networks: A survey. IEEE Commun. Surv. Tutor. **17**(4), 2317–2346 (2015)
70. Flauzac, O., González, C., Hachani, A., Nolot, F.: SDN based architecture for IoT and improvement of the Security. In: 2015 IEEE 29th International Conference on Advanced Information Networking and Applications Workshops, Gwangiu, pp. 688–693 (2015)
71. Scott-Hayward, S., OCallaghan, G., Sezer, S.: SSDN security: A survey. In: Proceedings of the IEEE SDN for Future Networks and Services, pp. 1–7 (2013)
72. Ojo, M., Adami, D., Giordano, S.: A SDN-IoT architecture with NFV implementation. In: In 2016 IEEE Globecom Workshops (GC Wkshps), pp. 1–6. IEEE (2016)
73. Berde, P., Gerola, M., Hart, J., Higuchi, Y., Kobayashi, M., Koide, T., Lantz, B., O'Connor, B., Radoslavov, P., Snow, W., Parulkar, G.: ONOS: Towards an open, distributed SDN OS. In: Proceedings of the Third Workshop on Hot Topics in Software Defined Networking, pp. 1–6, Chicago (2014)
74. Salman, O., Elhajj, I., Chehab, A., Kayssi, A.: Software defined IoT security framework. In: 2017 Fourth International Conference on Software Defined Systems (SDS), pp. 75–80. IEEE (2017)
75. Ziegler, S., Skarmeta, A., Bernal, J., Kim, E.E., Bianchi, S.: ANASTACIA: Advanced networked agents for security and trust assessment in CPS IoT architectures. In: 2017 Global Internet of Things Summit (GIoTS), pp. 1–6. IEEE (2017)
76. Rivera, D., de Oca, E. M., Mallouli, W., Cavalli, A., Vermeulen, B., Vucnik, M.: IoT Network monitoring and test of an industrial solution on Fed4Fire+ Platforms. In: The Proceedings of the 31th IFIP International Conference on Testig Software and Systems IFIP-ITCSS 2019. Paris, October 15–17, 2019 AU6
77. Casola, V., De Benedictis, A., Riccio, A., Rivera, D., Mallouli, W., de Oca, E.M.: A security monitoring system for Internet of Things. In: The Direct Science Journal, Internet of Things. Elsevier. https://doi.org/10.1016/j.iot.2019.100080
78. Mallouli, W., Wehbi, B., de Oca, E.M., Bourdelles, M.: Online network traffic security inspection using MMT tool. In: The 9th Workshop on System Testing and Validation (STV), Paris, France (October 2012)
79. Horcas, J.-M., Pinto, M., Fuentes, L., Mallouli, W., de Oca, E.M.: An approach for deploying and monitoring dynamic security policies. In: The Computers & Security Journal, Vol. 58, pp. 20–38, Impact Factor: 1.03 (May 2016). https://doi.org/10.1016/j.cose.2015.11.007
80. Choi, S.K., Yang, C.-H., Kwak, J.: System hardening and security monitoring for IoT devices to mitigate IoT Security vulnerabilities and threats. TIIS. **12**(2), 906–918 (2018)
81. d'Orazio, L., Lallet, J.: Semantic caching framework: An FPGA-based application for IoT Security monitoring. OJIOT. **4**(1), 150–157 (2018)
82. Kotenko, I.V., Saenko, I., Kushnerevich, A.: Parallel big data processing system for security monitoring in Internet of Things networks. JoWUA. **8**(4), 60–74 (2017)
83. Kotenko, I.V., Saenko, I., Branitskiy, A.: Framework for mobile Internet of Things Security monitoring based on big data processing and machine learning. IEEE Access. **6**, 72714–72723 (2018)
84. Singh, M.P., Chopra, A.K.: The Internet of Things and multiagent systems: Decentralized intelligence in distributed computing. ICDCS, 1738–1747 (2017)
85. Deri, L., Del Soldato, A.: An architecture for distributing and enforcing IoT Security at the network edge. In: iThings/GreenCom/CPSCom/SmartData 2018, pp. 211–218 (2018)
86. Deri, L., Del Soldato, A.: Enforcing Security in IoT and Home Networks. In: Proceedings of ITASEC 18 Conference (February 2018)
87. Zarca, A.M., et al.: Security management architecture for NFV/SDN-aware IoT Systems. IEEE Internet of Things J. https://doi.org/10.1109/JIOT.2019.2904123
88. IBM Analytics Whitepaper: IBM Point of View: Internet of Things Security. Online: https://www.ibm.com/downloads/cas/7DGG9VBO
89. Antonakakis, M., April, T., Bailey, M., Bernhard, M., Bursztein, E., Cochran, J., Zakir, D., Alex Halderman, J., Invernizzi, L., Kallitsis, M., Kumar, D., Lever, C., Ma, Z., Mason, J., Menscher, D., Seaman, C., Sullivan, N., Thomas, K., Zhou, Y.: Understanding the Mirai botnet, pp. 1093–1110. USENIX Security Symposium (2017)
90. Ramos, J.L.H., Bernabe, J.B., Skarmeta, A.F.: Managing context information for adaptive security in IoT environments. In: Proceedings of IEEE 29th International Conference on Advanced Information Networking and Applications Workshops, pp. 676–681 (Mar 2015)
91. Giannoni, F., Mancini, M., Marinelli, F.: Anomaly detection models for IoT time series data. CoRR abs/1812.00890 (2018)
92. Great Bay Survey: 71% of IoT Enterprise Security Professionals Not Monitoring IoT Devices In Real Time. Online: https://www.greatbaysoftware.com/great-bay-survey-71-of-iot-enterprise-security-professionals-not-monitoring-iot-devices-in-real-time/
93. CETIC's Embedded & Communication Systems: 6LoWPAN network analysis tool. Online: http://cetic.github.io/foren6/
94. Raza, S., Wallgren, L., Voigt, T.: SVELTE: Real-time intrusion detection in the Internet of Things. Ad Hoc Netw. **11**(8), 2661–2674 (2013)
95. Akshay, L., Perkins, E., Contu, R., Middleton, P.: Forecast: IoT Security, Worldwide, 2016. 07 April 2016; ID: G00302108; Online: https://www.gartner.com/en/documents/3277832
96. Galen, H.: Introducing Microsoft Azure Sphere: Secure and power the intelligent edge. Online: https://azure.microsoft.com/en-us/blog/introducing-microsoft-azure-sphere-secure-and-power-the-intelligent-edge/
97. Amazon Web Services: IoT Applications and Solutions. Website: https://aws.amazon.com/iot/
98. Scherb, C., Grewe, D., Wagner, M., Tschudin, C.F.: Resolution Strategies for Networking the IoT at the Edge via Named Functions, pp. 1–6. CCNC (2018)
99. Mai, H.L. et al.: Towards Content-Centric Control Plane Supporting Efficient Anomaly Detection Functions. To be published in CNSM 2019
100. Datta, S.K., Bonnet, C.: Next-generation, data centric and end-to-end IoT architecture based on microservices. In: ICCE-Asia 2018, 3rd International Conference on Consumer Electronics, June 24–26, 2018, Jeju, Korea
101. Castanheira, L., Parizotto, R., Schaeffer Filho, A.E.: FlowStalker: Comprehensive Traffic Flow Monitoring on the Data Plane using P4. ICC 2019: 1–6 (2019)
102. Chaabouni, N., Mosbah, M., Zemmari, A., Sauvignac, C., Faruki, P.: Network intrusion detection for IoT Security based on Learning techniques. IEEE Commun. Surv. Tutor., 1–1 (2019). https://doi.org/10.1109/COMST.2019.2896380

Ana Rosa Cavalli obtained her Doctorat d'Etat es Mathematics Science and Informatics from the University Paris VII in 1984. She was professor at TELECOM SudParis since 1990, director of the Software-Networks department and member of the research laboratory CNRS SAMOVAR. She is now emeritus professor at Institute Polytechnique de Paris. Her research interests include testing methodologies, monitoring, cybersecurity and resiliency.

Wissam Mallouli received his Master from Evry University in 2005 and his Ph.D. in computer science from Telecom SudParis in 2008. He is currently a research engineer at Montimage. His topics of interest cover formal testing and monitoring of functional and security behavior of distributed systems and ad-hoc networks. He is the R&D responsible in the MMT-IoT project.

Edgardo Montes de Oca graduated as engineer in 1985 from Paris XI University, Orsay, in electronics and computer science. He worked as research engineer in Alcatel's and Ericsson's research centers. He founded Montimage in 2004. His main interests are building critical systems using state-of-the-art fault-tolerance, testing and security techniques; and developing solutions with strong performance and security requirements.

Diego Rivera obtained his Ph.D. in 2017 from Télécom SudParis on a Quality of Experience framework and tool that considers business variables for the prediction of quality. His research interests include the evaluation of Quality of Experience in Services Over-The-Top, Operating Systems and Network Security. He worked for Montimage as research engineer and now recently as senior developer at Riscure.

Index Terms:

Author Queries

Chapter No.: 13 471536_1_En

Query Refs.	Details Required	Author's response
AU1	Please be aware that your name, affiliation and email address and if applicable those of your co-author(s) will be published as presented in this proof. If you want to make any changes, please correct the details now. Note that corrections after publication will no longer be possible. Please note that we standardly publish professional e-mail addresses, but not private ones even if it is provided in the manuscript. If you have a different preference regarding publication of your email address, please indicate this clearly on the proof. If no changes are required, please respond with "Ok".	
AU2	Please confirm if captured author group and ORCID ID is okay.	
AU3	Heading “Summary” has been changed to “Abstract”. Please confirm if okay.	
AU4	Please verify the hierarchy of heads as set.	
AU5	Please let us know where to insert the opening quote that is missing or please confirm whether to remove it.	
AU6	References 76 and 98 were identical and Reference 98 has been deleted. The subsequent references have been renumbered. Please check and confirm if appropriate.	
AU7	Photo is poor quality for the authors “Ana Rosa Cavalli”, “Wissam Mallouli”, “Edgardo Montes de Oca”, “Diego Rivera”, “Andreu Belsa Pellicer”. Please provide better quality photos if available.	

Note:
If you are using material from other works please make sure that you have obtained the necessary permission from the copyright holders and that references to the original publications are included.