IoT Security Training for System Developers: Methodology and Tools

Razvan Beuran, Jidong Wang, Min Zhao, Yasuo Tan

Japan Advanced Institute of Science and Technology

Abstract

Opportunities, as well as challenges, accompany the development of new technologies, and the Internet of Things (IoT) is no exception. While most companies tout the benefits of IoT, challenges are often overlooked. Thus, IoT devices come in a variety of shapes, from small sensors to home routers and factory equipment, each with specific characteristics. While many of us own IoT devices, some may not even recognize them as such, let alone be able to manage them. This lead to a series of significant security incidents, such as the much-publicized Mirai botnet distributed denial-of-service cyberattack. The solution is to develop safer and more secure IoT systems, and in this paper we discuss first the methodology needed to train the developers of such systems for this purpose. We then present two training platforms that we designed and implemented following this methodology: IoTrain-Sim, which is based on the Cooja network simulator, and IoTrain-Lab, which uses the FIT IoT-LAB testbed as infrastructure. The two platforms include training content in the form of tutorials and predefined scenarios, both for fundamental and security IoT training, that the trainees can follow to gain an in-depth understanding of IoT via hands-on practice. The evaluation we conducted from functionality, performance and user perspectives demonstrates that our systems have several advantages compared to other approaches in terms of learner support, availability, extensibility, flexibility and scalability.

Keywords: Internet of Things (IoT), security training, IoT simulation, Cooja network simulator, IoT testbeds, FIT IoT-LAB

1. Introduction

Since the advent of the Internet, the number of connected devices has been increasing steadily, and after the emergence of the Internet of Things (IoT) the growth of the number and variety of connected objects started to accelerate. Thus, according to [1], there were an estimated 12.3 billion connected IoT devices in the world in 2021, and it is predicted that this number will grow to around 27 billion by 2025.

With such a large pool of "available" devices, it is no wonder that various malicious actors have tried to put them to criminal use. One of the most publicized cyberattacks in recent times was the Distributed Denial of Service (DDoS) 2016 attack on the Dyn Domain Name System (DNS) provider in the US that used a botnet of IoT devices infected by the Mirai malware to generate traffic in excess of 1.2 Tbps; this is the largest DDoS attack to date, and resulted in the inaccessibility of several high-profile web sites, such as GitHub, Twitter, Reddit, Netflix, and Airbnb. Furthermore, the public distribution of the source code of Mirai lead to a proliferation of IoT based botnets that represent a major attack vector in today's Internet [2].

Malicious activities as the ones described above are made possible by the fact that IoT devices are vulnerable to various types of cyberattacks, even the most basic ones such as dictionary attacks, and many devices collect and store personal information in potentially insecure ways. While various solutions exist for the security aspects of IoT [3], a key requirement is that IoT developers need to have the necessary knowledge and skills to be able to implement those solutions. Obtaining such knowledge and skills is best achieved via security training that is integrated with the general technical training of professionals. This paradigm has been recently put forward by the Japanese government under the name "Plus Security," to signify that all employee job training should include security training *in addition* to professional training, with more security training needed for more technical jobs [4].

The main motivation of our work is to improve IoT security education and training, so that developers can gain the knowledge and skills that are needed for implementing secure IoT systems. For this purpose, we have (i) outlined a methodology regarding IoT security education and training, and (ii) designed and implemented two training tools based on this methodology: the simulation-based training system named IoTrain-Sim, and the testbed-based training system named IoTrain-Lab. These systems target both beginners and medium-to-advanced developers in the field of IoT, and provide a cost-effective and flexible solution for both fundamental and security training. Both systems were released as open-source software on GitHub, and already include various types of training content [5, 6]. The training content and system functionality can be modified or extended as needed.

Our approach provides a simple and inexpensive way for users to learn the basics of IoT technologies. It also ensures a safe environment for conducting hands-on IoT security training in realistic conditions, essentially being a cyber range for IoT. The guided learning that we include, based on tutorials and predefined scenarios, ensures that users with various skill levels can experience all the phases of security training, starting from forensics to attack and even defense training. This follows the typical structure of cybersecurity courses, such as the *Cybersecurity Attack and Defense Fundamentals Specialization* provided on Coursera, which is a series consisting of three courses: *Ethical Hacking Essentials*, *Network Defense Essentials*, and *Digital Forensics Essentials* [7].

This paper focuses on demonstrating the potential of using simulation and testbed environments for IoT security training, and our main contributions are:

- Discuss the general methodology for IoT security training that is the basis of our implementation, such as the approach we used for targeting different classes of learners from educational and training perspectives, the implementation requirements we formulated, etc.

- Present the IoTrain-Sim simulation-based and the IoTrain-Lab testbed-based training systems, emphasizing the characteristics of each approach and providing relevant security training examples for each of them.

- Evaluate the two systems we developed from functionality, performance and user perspectives, thus demonstrating their advantages in terms of learner support, availability, extensibility, etc.

The remainder of the paper is structured as follows. Section 2 presents several related work examples that are relevant to our approach. Section 3 discusses the motivation of our research and the methodology we propose for IoT security training. Sections 4 and 5 provide details about the design and implementation of IoTrain-Sim and IoTrain-Lab, respectively, and about the training content we have developed for each of them. The paper continues in Section 6 with the functionality, performance and user evaluation of the two systems, and ends with conclusions, acknowledgments and references.

2. Related Work

In this section we discuss existing IoT training programs and systems, and how they influenced the thought process behind our approach. The programs and systems are classified based on the type of activity provided, and we give several representative examples in each case. The categories we identified are:

- *Hands-on training exercises*: Training exercises developed in the academia for hands-on lab activities

- *Hands-on training programs*: Training programs provided by companies and institutions that have a hands-on aspect

- *Hardware-based training systems*: Training systems provided by companies as hardware-based solutions for training activities

- *Online courses*: Courses that are provided exclusively online and that have no hands-on aspects

Another type of training that has recently gained popularity is table-top training. In the area of IoT security, card games such as IoT-Poly [8] make it possible to learn how to conduct risk assessments for IoT systems. However, table-top exercises have neither a built-in learning component, nor hands-on aspects, so we consider them outside the scope of our analysis.

In what follows we shall discuss examples from each of the four categories we identified. An overview of the main characteristics of these systems is provided in Table 1, and a more detailed comparison with our work is presented as part of the functionality evaluation in Section 6.1.

Table 1: Overview of the related work examples

Name	**Type**	**Scope**
Smart Home Security Education	Hands-on exercises	Practice regarding smart home security cameras
IoT Security Exercises	Hands-on exercises	Practice regarding Web vulnerabilities of IoT devices
Watson IoT Academy	Hands-on online program	Practice regarding the IBM Watson IoT platform
IoT Fundamentals: IoT Security	Hands-on online course	Practice regarding defensive and offensive security
SEC556: IoT Penetration Testing	Hands-on online program	Practice regarding advanced offensive security
3 Rocks Technology IoT Training Systems	Hardware-based systems	Practice with training components such as gateway, camera, Arduino, etc.
STP IoT Training Systems	Hardware-based systems	Practice regarding home networks, healthcare, smart farms, smart logistics
IoT Training System for Smart Manufacturing	Hardware-based system	Practice with manufacturing components such as weight sensors, motors, conveyor belts
Cybersecurity and the Internet of Things	Online course	Security and privacy issues regarding programmed devices, connected homes, consumer wearables
Internet of Things: Business Implications and Opportunities	Online course	IoT-related business management aspects, such as the role of IoT in business strategy, etc.
IoT Security Fundamentals	Online course	Guidelines on the building and deployment of secure end-to-end IoT systems
IoT Security Education Framework	E-learning tool	Generating online e-learning materials and quizzes for IoT security education

2.1. Hands-on Training Exercises

Education activities are most effective when they also include interactive hands-on exercises, and various researchers in the academia have developed such training exercises. The systems presented below are closest in nature to our academic work, however the level of prerequisite knowledge they require is higher, making it difficult for beginner users to fully benefit of their content.

Smart Home Security Education. A set of hands-on lab exercises presented in [9] that allow students to conduct penetration testing against a set of smart home security cameras by using the Kali Linux distribution. The exercises are meant as a way for students to reveal and understand the vulnerabilities of real IoT devices. However, there is no learning component in this set of exercises, as students are expected to already have a good background on cyber attacks and Kali Linux before starting.

IoT Security Exercises. A set of exercises related to IoT security presented in [10] that are divided into two classes: basic and advanced. The basic exercises are aimed at teaching essential vulnerability detection and countermeasures by using WebGoat, which is a deliberately insecure application that allows developers to test common vulnerabilities in Java-based applications [11]. The advanced exercises employ more complex penetration testing techniques, such as using CSRF (Cross-Site Request Forgeries) to attack a digital signage application (emulated via a real Raspberry Pi device). The main purpose of the exercises is to familiarize developers and users with IoT device vulnerabilities, and they are deployed on CyExec, which is a virtual environment using VirtualBox and Docker technologies to create a low-cost cybersecurity exercise system.

2.2. Hands-on Training Programs

There are several companies and institutions that provide IoT training programs which include hands-on activities. While the practical aspect of these programs is very useful, their commercial nature makes it costly to attend them, and there is no freedom in extending their content (as is possible in the case of our work), which makes them less suitable for academic teaching.

Watson IoT Academy. An online program provided by IBM that includes hands-on practice with users' own hardware [12]. The content is tailored for the IBM Watson IoT Platform, which is a cloud-hosted service designed to simplify IoT device development and management [13]. The Watson IoT Academy training content is oriented towards professionals, and covers topics such as using IoT to connect business operations and improve operational efficiency, improvements in product development via IoT technologies, and sensor data collection and analysis using Watson. Introductory courses are provided at no cost and are self-paced, but most other courses are not free and are instructor-led.

IoT Fundamentals: IoT Security. A training course provided by Cisco Networking Academy [14]. The target skills for this course are: conducting end-to-end security assessments of IoT systems, gaining hands-on experience with IoT prototypes, deciding threat mitigation measures to minimize IoT solution risks, and becoming proficient in using real-world penetration and vulnerability testing tools. The course is instructor-led and has a duration of 50 hours.

SEC556: IoT Penetration Testing. An online program offered by the SANS Institute (Escal Institute of Advanced Technologies) that includes hands-on practice with hardware that is provided by SANS upon registration [15]. The program content is very technical and offensive-security oriented, covering topics such as assessing IoT network-facing controls and web applications, examining IoT hardware to discover functionality and use interaction points to obtain data, uncovering firmware from hardware and exploring it for secrets and implementation issues, sniffing and interacting with WiFi, LoRA, and Zigbee wireless technologies, etc.

2.3. Hardware-Based Training Systems

Other companies focus on providing IoT training systems, which are dedicated hardware platforms that can be used for IoT security training. The main advantage of hardware systems is that they make it possible for learners to practice with real hardware; however, they suffer from ease of deployment and scalability issues, and these are the particular aspects that we improved on via our simulation and testbed-based approaches.

3 Rocks Technology IoT Training Systems. Training systems by 3 Rocks Technology, an engineering training system provider that currently offers two solutions for IoT training, called Embedded IoT Training System (IOT-16300) and Internet of Things Trainer (IOT-16900) [16]. Both are hardware-based IoT training systems that integrate various hardware, such as a Virtual Private Network (VPN) Smart Gateway, a Web camera, an Arduino board, a Raspberry Pi module, audio and sensor modules, depending on the configuration. In terms of training content, 3 Rocks Technology provides a series of scenarios, such as home appliance control and IoT network construction, IoT farm management, IoT fire alarm, IoT controlled toys, IoT security application, and so on, that the trainees can use to understand how to design, implement, and operate an IoT system.

STP IoT Training Systems. A series of more than a dozen training systems provided by the Scientific & Technical Products (STP) company, covering areas such as home networks, healthcare, smart farms, and smart logistics [17]. A representative model is IoT-1000, which is a Raspberry Pi 3 based system that makes it possible to conduct hands-on practice with various sensors and actuators, setting up an IoT server, saving sensor and actuator data into a database, real-time streaming of a camera feed, using sensors and actuators via WiFi and Bluetooth communication, as well as conducting experiments in different scenarios by following the provided textbook.

IoT Training System for Smart Manufacturing. A training system developed by National Kaohsiung University of Science and Technology and Asia University in Taiwan. This system too is based on a Raspberry Pi board, and it includes several components specific to manufacturing, such as a weight sensor, servo motors and stepping motors, as well as conveyor belts [18]. This IoT training system lets students follow the process of implementing a mock-up industrial system that is monitored via a Web/cloud application, and lets them have an industry-related real-machine operating experience.

2.4. Online Courses

Another approach to IoT and IoT security education is to provide online courses that include relevant information on these topics. While this makes it possible for learners to accumulate knowledge about those topics, the lack of a practical hands-on aspect prevents skill development, something that we took into account when including both theoretical knowledge (tutorials) and practical exercises in our training content.

Cybersecurity and the Internet of Things. An online course available on Coursera that is implemented by the University System of Georgia [19]. The course explores security and privacy issues regarding programmed devices, connected homes, consumer wearables, and so on, via reading materials, videos, case studies, and quizzes. The course is instructor-led and has a duration of approximately 11 hours.

Internet of Things: Business Implications and Opportunities. An online course offered by GetSmarter that is implemented by Massachusetts Institute of Technology (MIT) [20]. This course takes a business management rather than a technical perspective on IoT, addressing topics such as considering the role of IoT in business strategy, learning about the technological components that support IoT, and integrating Information Technology (IT) and IoT technologies with business strategy and operations. This course too is instructor-led and has a duration of 36 to 48 hours.

IoT Security Fundamentals. An online course provided by Telecoms & Tech Academy [21]. This course focuses on teaching how to examine IoT device vulnerabilities, understanding how these vulnerabilities should be addressed and mitigated, how to secure IoT products and services, how to build and deploy secure IoT solutions, and how to examine end-to-end IoT security issues. The course can take place either on-demand or as a live online classroom.

IoT Security Education Framework. A tool for generating web-based 3D and 360° virtual reality e-learning materials and quizzes for IoT security education that was presented in [22]. The automatic generation of the content helps educators in creating materials for different kind of scenarios. However, this framework has no practical training aspect, hence it can only be used for knowledge training and not skill training; this matches the fact that the framework targets mainly regular IoT users, not developers.

3. IoT Training Methodology

In this section we outline our methodology for IoT training, including by formulating a set of implementation requirements. We also provide information about the external components used in the systems we developed.

3.1. Methodology Overview

The main viewpoints regarding security in IoT systems are: (i) the developer perspective, (ii) the administrator perspective and (iii) the user perspective. Our methodology targets mainly the developers who implement IoT systems, but administrators that manage systems integrating IoT devices and regular IoT system users may also benefit of the technical and non-technical information included in the training content, respectively.

The review of representative IoT security training approaches that we did in Section 2 emphasizes the differences between the four categories we identified. Thus, hands-on exercises are a good opportunity to practice hands-on skills in an academic environment. Hands-on programs represent a complete solution, as they combine instructor-led theoretical training with hands-on practice to provide both knowledge and skill learning attributes. These two approaches were the main source of inspiration for our methodology.

As for training systems, they serve as hands-on practice platforms and are accompanied by learning materials, usually in the form of textbooks, that facilitate the training activity; their main disadvantage, however, is that the need to purchase the actual system can become a barrier to entry for potential learners, both in terms of cost and availability. Lastly, online courses only provide knowledge to learners, hence lack the hands-on component that is critical for IoT and IoT security training, so we consider it to be the least effective approach.

Our analysis of existing systems guided the design of a methodology that provides the mechanisms for acquiring both in-depth knowledge and practical skills regarding IoT in a scalable, flexible and extensible manner. First of all, we identified three classes of trainees:

1. *Beginners*, who have no technical knowledge about IoT and IoT security, hence need to learn everything from scratch
2. *Intermediate users*, who have some understanding of IoT and IoT security, but have limited skills to address real-life situations due to a lack of systematic training
3. *Advanced users*, who have knowledge about IoT and IoT security and programming ability, but need to study more in order to cover new areas, improve their practical skills, etc.

Each of these three classes of learners has its own specificities regarding the most suitable training method, so the next step was to decide *how* to train those users. The main issues in this context are: (i) what is the educational component of the training, (ii) what is the practical component of the training, (iii) how to ensure the effectiveness of the training, and (iv) how to actually conduct the training, as it will be discussed next.

Educational Aspect. For the educational aspect, given that tutorials are in general widely used as self-study materials, for all levels of users it is essential to make available such tutorials, and this was the first principle that lead our training content development process [23].

Practical Aspect. The other issue is the practical training aspect of IoT-related education, which is an essential component of our methodology. Beginners may follow tutorials initially to acquire basic knowledge, but once their level increases, and also for intermediate users, this is not sufficient anymore. Predefined scenarios are the best way to gain a practical insight into the way IoT devices operate and communicate, what are the corresponding security concerns, and so on. Therefore, as a second content development principle, we decided to create various scenarios targeted at intermediate to advanced users.

Learning–Viewing–Doing Paradigm. As the level of trainees increases even further, and also for advanced users, only viewing predefined scenarios is not adequate anymore, and trainees should be tasked with modifying them, for instance to reproduce security attacks and even to design defense mechanisms. The advantages of learning-by-doing versus learning-by-viewing have been demonstrated both theoretically and practically by many research works, such as the study on data analyst productivity presented in [24]. Consequently, the IoT training methodology that we envisioned follows a Learning–Viewing–Doing paradigm, as follows: (i) learning via tutorials, (ii) viewing predefined scenarios, and (iii) doing hands-on experimentation.

Training Infrastructure. In order to support the practical aspect of our methodology, we aimed for an approach that makes it possible to conduct practical hands-on training in a scalable manner. Often times hardware platforms such as Raspberry Pi and Arduino are used to teach IoT. While this is a good starting point, the time and effort required to set them up, especially for large classes of students and large training scenarios, is a clear disadvantage. Therefore, we decided to proceed with a different approach: leverage existing simulation systems to make possible realistic training with low setup costs even for large-scale training. Since simulation does lack the realism of an actual hardware system, we also employed an existing IoT testbed for IoT training purposes (see Section 3.3 for the actual external components used).

3.2. Implementation Requirements

While the aforementioned Learning–Viewing–Doing paradigm is the basis of our IoT training system design, we have also formulated a set of more concrete requirements that we believe are important for the implementation of an effective and flexible IoT training system:

1. *Multi level*: Support conducting training for users with different levels of knowledge and skills regarding IoT
2. *Content rich*: Make available different types of training content and training modes that cover the key aspects IoT technologies

3. *Open source*: Make possible training content and system customization for various categories of trainees
4. *Low cost*: Minimize costs related to the training, both in terms of training infrastructure and training content
5. *Multi user*: Achieve scalability through supporting multiple trainees simultaneously
6. *Easy management*: Simplify content addition and modification via a rigorous structure, both for the training content and the system itself

The criteria 1 and 2 above ensure that the system can support a wide range of users, criteria 3 and 4 are related to the accessibility of the system, and criteria 5 and 6 refer to the fact that the system should be easy to administer. As part of the functionality evaluation in Section 6.1, we will discuss in more detail how we addressed each of these requirements in our implementation.

3.3. External Components

To achieve our training methodology goals while minimizing development costs, our system makes use of several external components that it leverages in order to provide IoT hands-on training capabilities. In what follows, we introduce each of them briefly: Contiki and Cooja, which are used in IoTrain-Sim, and the Future Internet Testing (FIT) facility IoT-LAB testbed, which is the infrastructure for IoTrain-Lab.

Contiki OS. An open-source operating system (OS) for IoT that connects low-cost, low-power microcontrollers to the Internet [25]. Contiki is a powerful toolbox for building complex wireless systems, and it supports fully standard IPv6 and IPv4, along with low-power wireless standards and technologies, such as 6LoWPAN (IPv6 over Low-Power Wireless Personal Area Networks), RPL (IPv6 Routing Protocol for Low power and Lossy Networks), and CoAP (Constrained Application Protocol).

While there are several other operating systems for IoT devices, we have selected Contiki OS as a representative operating system, especially for mesh networks and Wireless Sensor Networks (WSNs). In addition to being an open-source OS, another advantage of Contiki OS is that its applications are written in the standard C language. Moreover, there are many examples in the Contiki source code tree to help users start coding. Note that development of a new Contiki version, called Contiki-NG, has started since we released our system, and we are considering to support it as well in the future.

Cooja Network Simulator. An extensible network simulator included with the Contiki OS source code that is capable of emulating Tmote Sky, Z1, and other Contiki nodes. Cooja allows developers to run their applications over large-scale wireless networks with extreme detail for the fully-emulated hardware devices. These emulation capabilities make Cooja an ideal solution for realistic IoT experiments and training at low cost, since no hardware needs to be purchased.

The code to be executed by a node when using Cooja is the same with the firmware that will be uploaded to the physical node. Interactions with the simulated nodes are performed via plugins, such as `Simulation visualizer`, `Collect-view` and `Radio logger`. Simulation scenarios can be stored in XML files with the extension CSC (Cooja Simulation Configuration) that contain information about the simulation environment, plugins, nodes and their positions, radio communication medium, etc.

FIT IoT-LAB. A large-scale IoT testbed that is part of the FIT facility in France [26, 27]. FIT IoT-LAB is suitable for experimenting with actual wireless sensor devices and other heterogeneous communicating objects. It supports different communication technologies, such as IEEE 802.15.4, Sub-GHz Industrial Scientific and Medical (ISM) bands, Bluetooth Low Energy (BLE), and the long-range radio communication technique LoRa. The testbed also has a large choice of hardware boards (23 in total, including Arduino, Microchip, Nordic, and the custom M3 and A8-M3 boards), multiple operating systems (RIOT, Contiki, FreeRTOS, etc.) and different topologies at 9 physical sites, for a total of 1500+ nodes.

The open-source, open-access and multi-user nature of the FIT IoT-LAB testbed, as well as the fact that it can be used remotely from anywhere, make it very suitable as infrastructure for IoT training purposes.

4. IoTrain-Sim

In this section we present the design and implementation of IoTrain-Sim, provide an overview of the available training content, and discuss an IoT security training example.

4.1. System Overview

The functionality of IoTrain-Sim is related to three main aspects: (i) handle the IoT training content; (ii) interface with the Cooja network simulator; (iii) interact with the trainees.

The overall architecture corresponding to this functionality is shown in Figure 1. Instructors create training content that is placed in the content database; as mentioned already, we have already developed such training content, but additional content can be created if needed. The core functions of IoTrain-Sim are to retrieve training content from the database and to display it to trainees, either in the form of tutorials, or by driving the Cooja simulator in accordance with the included simulation scenarios. Learners are able to control the training process via a graphical user interface (GUI) or a command-line interface (CLI).

This simulator-based design presents several advantages for IoT security training, as follows:

- The simulation environment provides a safe way to conduct security training in which malicious nodes are used and dangerous traffic is generated.

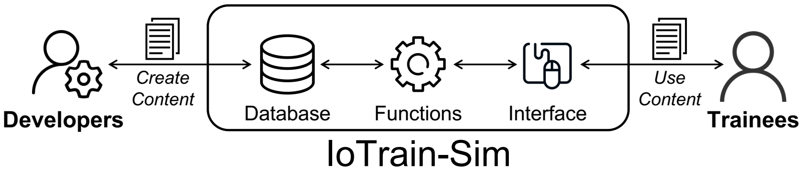

Figure 1: Overview of the IoTrain-Sim architecture.

- Cooja emulates actual hardware behavior, hence the knowledge and skills gained via such training are readily applicable to real devices.
- Large-scale experiments can be easily conducted, thus providing a learning environment that cannot be built easily via actual hardware setups.

The core functions of IoTrain-Sim are implemented using the Python programming language, and include:

1. Configuring the system upon start, including configuration changes to Contiki OS that are conducted before the training starts, and reverted back automatically once the training ends.
2. Displaying the user interface that provides functionality for opening on request tutorials in PDF format (via the `webbrowser` Python library) and Cooja simulation files (via a custom class driving the execution of Cooja).

The IoTrain-Sim user interface is currently available in both GUI and CLI modes, with trainees being presented activity choices as pictured in the GUI screenshot in Figure 2. The menus that are shown to trainees are constructed based on the content structure representation that will be discussed in Section 4.2.

4.2. Training Content

Training content for IoTrain-Sim consists of three types of files: (i) PDF files for tutorials; (ii) CSC files for representing predefined Cooja simulation scenarios; (iii) C language files for modifying the behavior of the simulated Contiki nodes. These files are stored in a structured hierarchy of directories. For each activity, the directory `Instruction/` contains tutorial-related files, and the directory named `Simulation/` includes the simulation-related resources. This straightforward organization makes it easy for instructors to add or edit the files as needed, and also simplifies content sharing.

4.2.1. Content Overview

The training content structure is represented internally as a multi-level ordered Python dictionary (see Figure 3). By editing this file, instructors can decide what content is made available to trainees, how it is organized, and so

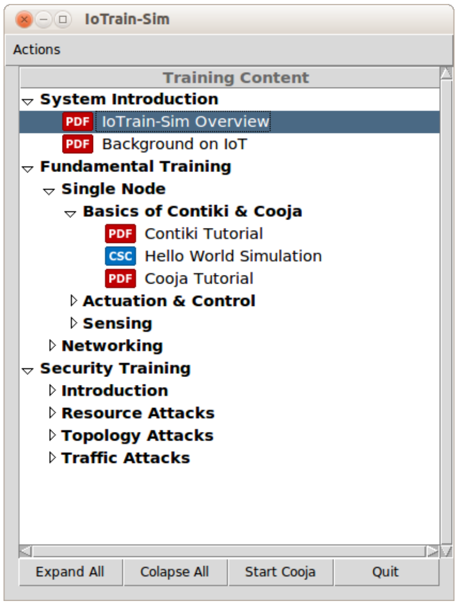

Figure 2: Screenshot of the IoTrain-Sim GUI.

on. To simplify the representation, only the content file names need to be included (without any path information), as IoTrain-Sim will automatically locate the indicated files in the directory hierarchy.

While our system is fully extensible from a training content point of view, we have already designed and included with IoTrain-Sim a relatively thorough set of training content that allows even inexperienced users to acquire all the basic knowledge needed in order to learn IoT security skills. The training content is organized in three categories, as detailed next (see also Figure 4).

System Introduction. Aimed at all user levels, the system introduction tutorial covers the background of IoT and IoT security, and also IoTrain-Sim itself.

Fundamental Training. Aimed at users with beginner to intermediate knowledge about IoT, the fundamental training starts with the basics of Contiki OS and Cooja. Then tutorials and hands-on exercises are provided on the elementary operation of IoT devices. Various Contiki-based IoT devices are introduced, such as actuators, controllers, and sensors, to allow users to master the basic operation of single nodes first. Following that, communication techniques are explained, with practical examples on how to do broadcast, etc.

Security Training. Aimed at users with intermediate to advanced knowledge about IoT, the security training content builds upon the fundamental training content to provide IoT security knowledge. The focus is on the security of RPL,

```
training_content = OrderedDict([
 (‘System Introduction’, OrderedDict([
  (‘IoTrain-Sim Overview’, ‘iotrain-sim_overview.pdf’),
  (‘Background on IoT’, ‘background_iot.pdf’)
 ])),
 (‘Fundamental Training’, OrderedDict([
  (‘Single Node’, OrderedDict([
   (‘Basics of Contiki & Cooja’, OrderedDict([
    (‘Contiki Tutorial’, ‘contiki_tutorial.pdf’),
    (‘Hello World Simulation’, ‘hello-world.csc’),
    (‘Cooja Tutorial’, ‘cooja_tutorial.pdf’)
   ])),
   (‘Actuation & Control’, OrderedDict([
    (‘Overview’, ‘actuation_control_overview.pdf’),
    (‘LED Tutorial’, ‘led_tutorial.pdf’),
    (‘LED Simulation’, ‘led.csc’),
    (‘Button Tutorial’, ‘button_tutorial.pdf’),
    (‘Button Simulation’, ‘button.csc’),
    (‘Timer Tutorial’, ‘timer_tutorial.pdf’),
    (‘Timer Simulation’, ‘timer.csc’)
   ])),
   (‘Sensing’, OrderedDict([
    (‘Sensor Tutorial’, ‘sensor_tutorial.pdf’),
    (‘Sensor Simulation’, ‘sensor.csc’)
   ]))
  ])),
  (‘Networking’, OrderedDict([
   (‘Communication’, OrderedDict([
    (‘Broadcast Tutorial’, ‘broadcast_tutorial.pdf’),
    (‘Broadcast Simulation’, ‘broadcast.csc’)
   ]))
  ]))
 ])),
 ...
])
```

Figure 3: IoTrain-Sim content structure representation as a hierarchical Python ordered dictionary (excerpt from the actual file).

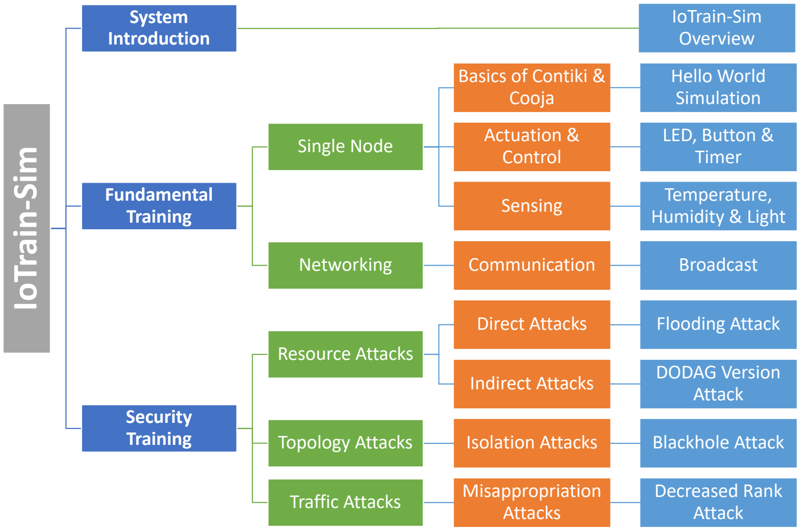

Figure 4: Structure of the IoTrain-Sim training content.

an open standard routing protocol for low-power and lossy networks that is often used in IoT/WSN deployments. This was motivated by the fact that, when deployed in complex environments, WSNs are especially vulnerable to routing attacks [28]. More specifically, our security training content is organized according to the taxonomy of attacks against the RPL protocol presented in [29], as detailed below:

- Resource attacks: Attacks that aim to exhaust node resources (processing load, memory, power consumption) by forcing the legitimate nodes to perform unnecessary actions. In direct attacks malicious nodes interfere directly with network operation, whereas in indirect attacks they disturb legitimate node operation.
- Topology attacks: Attacks against the network topology that aim to disrupt the normal network organization, for example by isolating some nodes or by causing non-optimal routing to be performed.
- Traffic attacks: Attacks that attempt to introduce malicious nodes into the network for purposes such as eavesdropping on the traffic or impersonating legitimate nodes.

Each class of attacks is illustrated via examples, such as the flooding attack for the direct resource attack class, and the blackhole attack for the topology attack class (see the lower half of Figure 4). For each attack, we have created a tutorial that explains the theoretical background and practical elements of the attack mechanisms, and included an implementation of the attack, so that trainees can reproduce it in IoTrain-Sim, as it will be explained in Section 4.3. Moreover, an explanation about the way to defend from the attack is also provided, so that trainees can also develop defense skills against that attack.

4.2.2. Content Creation

For clarity purposes, we briefly describe below the procedure needed in order to add new training content into IoTrain-Sim:

1. Create and add specific files to the training database, depending on the file type and purpose (PDF for tutorials, CSC and C for Contiki/Cooja).
2. Register the new files with the system by updating the database structure representation in the file `contents.py` shown in Figure 3.

To produce tutorials, developers simply need to export the learning materials they create to PDF format, and store these PDF files in the training database. As for the simulation-related files, the creation procedure is outlined below:

- *Fundamental Training Simulations*: To implement a fundamental training simulation, one has to first write a Contiki OS application using C language, then import the application into Cooja, and select an appropriate hardware platform for compiling and generating the simulation. The result should be saved in the training database as a CSC file, so that trainees can simply open the simulation through the Cooja interface.

- *Security Training Simulations*: Our approach to security training is to have two simulations per training topic, a *reference simulation* that contains the normal conditions for a scenario, and an *attack simulation* that introduces malicious nodes in that scenario. Trainees run both these simulations, then use Cooja to visualize the simulation conditions and investigate issues related to the attack scenario, such as identifying the malicious nodes, determining the effects of the attack, and so on. Intermediate users can view these simulations to gain insight into the attack, while advanced trainees can be tasked with modifying the Contiki source code to implement attacks by themselves under the guidance of tutorials, or even with designing defense mechanisms against those attacks.

4.3. Security Training Example

To illustrate how security training is conducted using IoTrain-Sim, we provide an example based on one of the most typical attacks on the RPL protocol, the flooding attack [30]. In this attack malicious nodes send network management packets to all the neighbors in their transmission range, forcing the victim nodes to update their internal data and reply with other management packets, thus flooding the network with unnecessary traffic. The effects of the attack can be easily understood by comparing the reference and attack simulations, as it will be explained next. Screenshots for the reference and attack simulations are shown in Figures 5 and 6, respectively.

The main points regarding the reference simulation in Figure 5 are as follows (see the left-hand side window named “Network”):

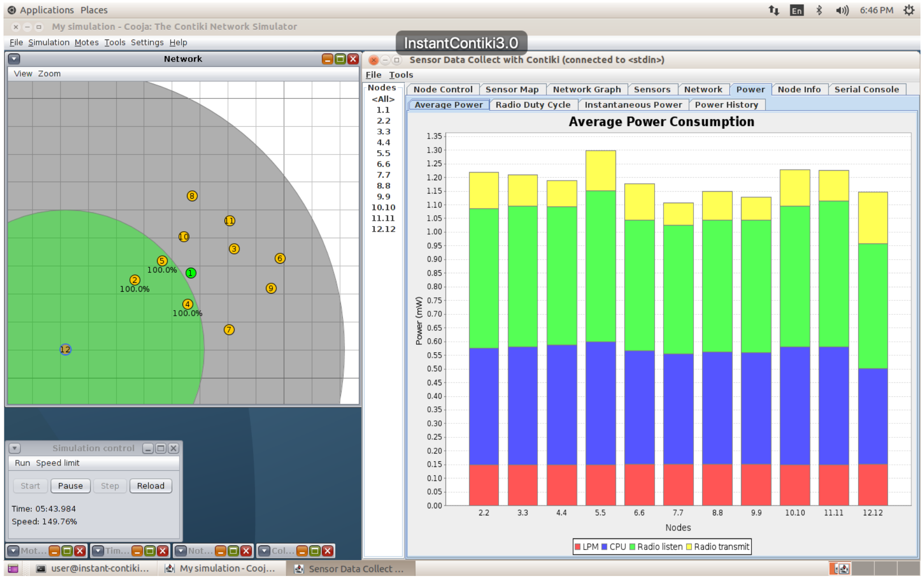

Figure 5: Screenshot of the reference simulation scenario.

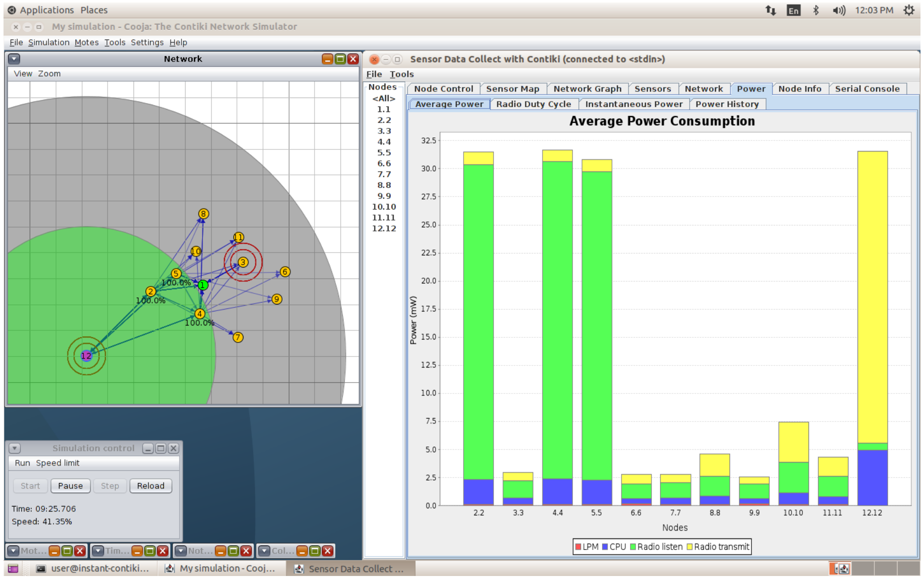

Figure 6: Screenshot of the attack simulation scenario.

- Node 1, which is shown in green color, is a sink node that acts as a border router.
- Nodes 2 through 12, which are shown in yellow color, are sender nodes that act as sensors.
- Nodes 2, 4 and 5 are in the range of node 12, shown as a green area, hence they can communicate directly.
- All sender nodes have nearly the same measured average power consumption (see the right-hand side window).

When comparing the reference simulation with the attack simulation shown in Figure 6, in which node 12 was replaced by a malicious node, we note the following (see the right-hand side window named "Average Power Consumption"):

- Nodes 2, 4 and 5, which are in the range of node 12, have a very high power consumption, and the power consumption for the other sender nodes is also higher than before.
- For the malicious node 12, the radio transmit operation (the yellow area of the bar graph) represents a large proportion of the power consumption, since it continuously sends messages to neighboring nodes.
- For nodes 2, 4, and 5, the radio listen operation (the green area of the bar graph) uses significant power, since these nodes continuously receive messages from node 12.

Through this kind of comparative analysis, intermediate users can conduct forensics-like investigations to identify the malicious nodes (i.e., node 12 in our example), and recognize the effects of various types of attacks (such as increased power consumption for nodes 2, 4, 5 and 12 in the example we provided). Advanced users can design or modify malicious node behavior to produce new attacks or change existing ones, thus gaining insight into IoT attack mechanisms. In the next stage, this insight can be used to design and implement defense mechanisms against those attacks, and the effectiveness of such mechanisms can be validated through repeated experiments. Since these activities are conducted via Cooja simulations, they are both safe to perform and realistic.

5. IoTrain-Lab

In this section we introduce the design and implementation of IoTrain-Lab, give an overview of the available training content, and discuss an IoT security training example.

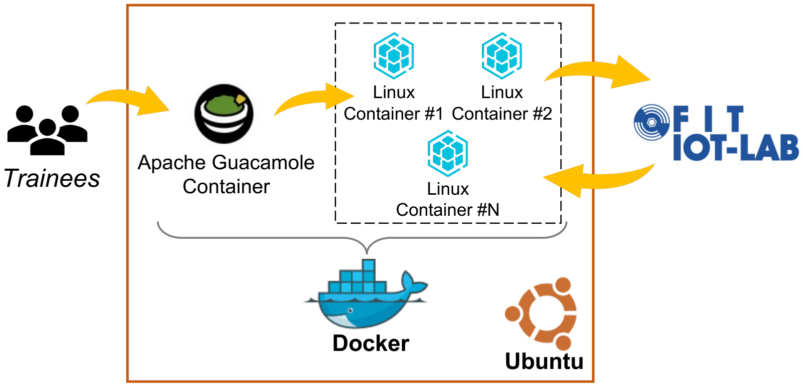

Figure 7: Overview of the IoTrain-Lab architecture.

5.1. System Overview

The functionality of IoTrain-Lab is related to two main aspects: (i) manage the trainee access to the training platform; (ii) interface between the training platform and the FIT IoT-LAB testbed.

The overall architecture corresponding to this functionality is shown in Figure 7. In order to ensure that multiple trainees can access the training platform simultaneously, we employed container technology to create containers in which all the necessary tools are already installed. In particular, we used Docker to deploy lightweight CentOS Linux containers that use Xfce as desktop environment. Linux container access management is done via another container that runs the Apache Guacamole clientless remote desktop gateway. All the settings needed to access the FIT IoT-LAB testbed are set up in the Linux container in advance by the course instructor/administrator, thus making it possible for trainees to begin the training just by logging in into those containers.

This testbed-based design has several advantages for IoT security training:

- The testbed environment provides a safe way to conduct security training in which malicious nodes are used and dangerous traffic is generated.
- The testbed is made up of real IoT devices, hence the knowledge and skills gained via such training are readily applicable in real life.
- Large-scale experiments with actual hardware can be conducted, thus providing a learning environment that cannot be built easily via local setups.

5.2. Training Content

While IoTrain-Lab is fully extensible from a training content point of view, we have already included a set of tutorials that allows even inexperienced users to acquire basic IoT knowledge and skills. The training content is organized in two categories, as detailed next (see also Figure 8).

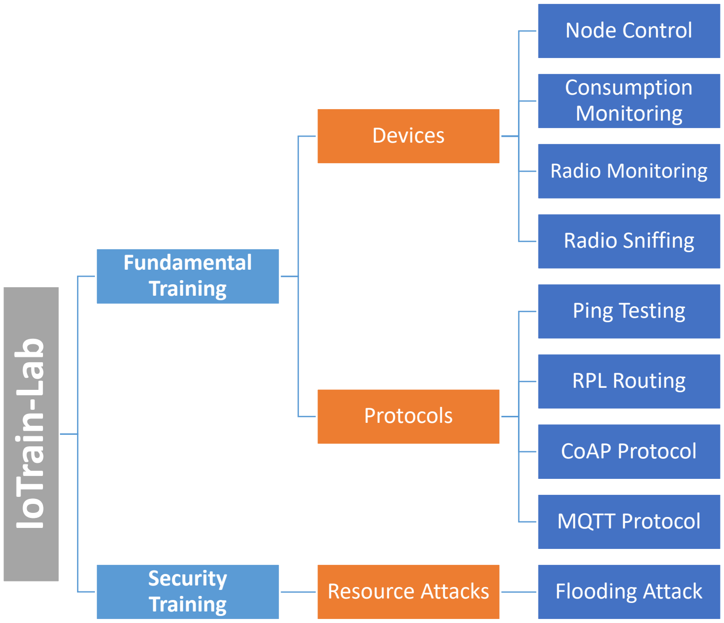

Figure 8: Structure of the IoTrain-Lab training content.

Fundamental Training. Aimed at users with beginner to intermediate knowledge about IoT, the fundamental training contains tutorials about how to control and monitor the nodes, and also about radio sniffing. A series of tutorials about network protocols is included as well, starting with basic ones such as RPL, and continuing with CoAP and MQTT tutorials for more experienced users.

Security Training. Aimed at users with intermediate to advanced knowledge about IoT, the security training content builds upon the fundamental training content to provide IoT security knowledge. The current focus is on the security of RPL, and the included tutorial is based on a similar tutorial in IoTrain-Sim that targets the flooding attack, which is an attack targeting network resources. Via this tutorial trainees are able to experiment with the flooding attack while using real IoT devices. Introducing additional attack examples is planned for the future.

All of the above-mentioned tutorials are available on the GitHub site of IoTrain-Lab [6], with detailed steps on how to conduct each training exercise, as well as relevant data and screenshots. We have labeled each exercise with its difficulty level, and they are presented in an increasing order of difficulty, thus making possible for trainees to gradually learn more and more advanced techniques.

5.3. Security Training Example

The IoT security training example we include here refers to the flooding attack on network resources, in which a large number of packets, typically "HELLO" messages, are transmitted unnecessarily in order to overwhelm the network.

This exercise uses four M3 nodes on the FIT IoT-LAB testbed to conduct a sequence of experiments via which the normal and attack network conditions can be compared. To implement the attack, the source code of the RPL protocol must be modified in order to introduce malicious node behavior. The tutorial instructs trainees to run the reference experiment first, save the power consumption data, then modify the source code and run the experiment again. By comparing the power consumption data for the reference and attack scenarios it is possible to examine the influence the attack had on the nodes.

In Figure 9 we show the experiment results regarding the effect the flooding attack had on power consumption. The measurements are conducted in a User Datagram Protocol (UDP) server-client scenario, and the results shown are measured by the UDP server in this scenario. The key insights are as follows:

- In the normal case, which is shown with dotted blue line as reference, the power consumption varies periodically according to the transmission pattern, without any significant spikes.
- In the flooding attack case, which is shown with continuous red line, we note strong fluctuations in the power consumption, as the operation pattern is completely altered, thus indicating that an attack is taking place.

Learners can conduct such an analysis of the differences in power consumption patterns between the reference and attack scenarios in order to understand the symptoms of the attack, and even to assess the effectiveness of the defense mechanisms they could implement against a given attack.

6. System Evaluation

This section discusses the evaluation of IoTrain-Sim and IoTrain-Lab, first from a functionality perspective, then regarding the performance characteristics of IoTrain-Sim, and the user evaluation of IoTrain-Lab.

6.1. Functionality Evaluation

For the functionality evaluation, we compare IoTrain-Sim and IoTrain-Lab with the two examples of hands-on training exercises introduced in Section 2.1, *Smart Home Security Education* and *IoT Security Exercises*, which are the closest to our work. In addition, we compare our systems with one example from each of the IoT training approaches presented in Sections 2.2 and 2.3: *Watson IoT Academy* by IBM as a representative of hands-on training programs, and the *3 Rocks Technology Systems* as representative of the hardware-based training systems category. Due to the nature of our own systems, we have excluded the online courses from our comparison, as they lack hands-on practice.

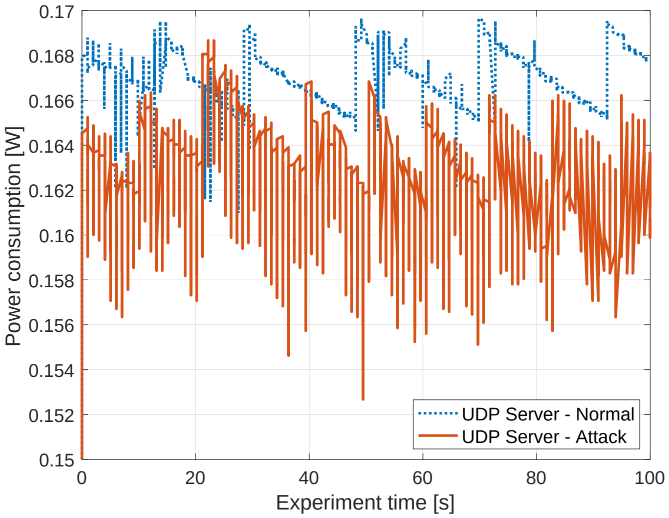

Figure 9: UDP server power consumption for the normal and flooding attack scenarios.

Feature Comparison. Table 2 shows a comparison between the two IoT training systems that we developed and: *Smart Home Security Education* [9] (shortened to the *Smart Home* in the table header), *IoT Security Exercises* [10] (shortened to *IoT Security*), *Watson IoT Academy* [12] (shortened to *Watson IoT*) and *3 Rocks Technology Systems* (shortened to *3 Rocks*) [16]; see Section 2 for details on each of them.

The upper part of the table compares general features, and emphasizes that the advantages of IoTrain-Sim and IoTrain-Lab are the wider target audience and lower prerequisite knowledge, as well as their free cost. Their availability as free downloads on GitHub also distinguishes them from other academic projects that are not publicly available. The highest cost is associated with the *3 Rocks Technology Systems* because of the need to purchase custom hardware platforms, which may deter individual learners.

The middle part of the table focuses on training content, showing that all the presented training systems include practical hands-on training and have a relatively good content coverage. The two academic projects, however, have a more narrow focus (penetration testing), and do not include any learning content, except for ethics topics in the case of *IoT Security Exercises.* They are also instructor led for the students enrolled in those institutions; *Watson IoT Academy* represents a special case in this category by providing some instructor-led courses to the public, of course at a cost.

The lower part of the table emphasizes the capabilities of each platform, showing that the multitude of supported boards of the FIT IoT-LAB testbed used by IoTrain-Lab provides most diversity in training, whereas the limitations

Table 2: Feature comparison of IoTrain-Sim and IoTrain-lab with related work

System / *Feature*	**IoTrain-Sim**	**IoTrain-Lab**	**Smart Home** [9]	**IoT Security** [10]	**Watson IoT** [12]	**3 Rocks** [16]
Target audience	Students and professionals	Students and professionals	Students	Students and professionals	Professionals	Students
Prerequisite knowledge	Low	Low to medium	Medium to high	Medium	Medium	Medium to high
Availability	Free download	Free download	Not public	Not public	Online learning	Equipment purchase
Registration requirement	No	No (Only for FIT IoT-LAB)	Not applicable	Not applicable	Yes	No
Cost	Free	Free	Not applicable	Not applicable	Medium	High
Connectivity	Offline	Online	LAN	LAN	Online	LAN
Training materials	Tutorials and simulations	Experiment tutorials	Not provided	Only ethics education	Videos, simulations, documents	IoT application examples
Training coverage	Fundamental and security training	Fundamental and security training	Pentesting	Pentesting (basic & advanced)	IBM Watson IoT platform related	Design and implementation principles
Training method	Self paced	Self paced	Instructor led	Instructor led	Self paced (with or without labs), instructor led	Self paced
Device type	Virtual	Real	Real	Virtual/real	Virtual/real	Real
Available devices	3 kinds of sensors, 1 kind of actuator	13 kinds of boards	4 kinds of security camera, smart phone	Generic, Raspberry Pi	Same with IBM Watson IoT Platform	8 kinds of sensors, 7 kinds of actuators
Device mobility	No	Yes	No	No	No	No
WSN capabilities	Yes	Yes	No	No	No	No
Supported OSs	Contiki OS	FreeRTOS, RIOT, Contiki-NG, etc.	Not available	Raspbian, Ubuntu	OS independent	Debian, ARM, RISC OS

Table 3: Assessment from an implementation requirement perspective

System Requirement	IoTrain-Sim	IoTrain-Lab	Smart Home	IoT Security	Watson IoT	3 Rocks
Multi level	○	○		○	○	
Content rich					○	○
Open source	○	○				
Low cost	○	○	○	○		
Multi user		○		○	○	
Easy mgmt.	○	○			○	

of the Cooja network simulator mean that IoTrain-Sim has less capabilities. *Smart Home Security Education* provides several real security cameras, but *IoT Security Exercises* is very limited in terms of device availability. One notable aspect in this category is the device mobility support available in IoTrain-Lab via the FIT IoT-LAB mobile robots, and the support of WSN training for both our systems, owing to their ability to work with a large number of simulated or real devices.

Implementation Requirements Assessment. In Table 3 we present the results of the assessment we conducted regarding the system implementation requirements formulated in Section 3.2.

Regarding the multi-level nature, both IoTrain-Sim and IoTrain-Lab have this feature by design, which is also the case for *IoT Security Exercises* and *Watson IoT Academy*. On the other hand this feature is not included in *Smart Home Security Education*, and the *3 Rocks Technology Systems* have a custom-hardware nature that limits their applicability to a wide range of users.

As for content availability, IoTrain-Sim and IoTrain-Lab are relatively new systems, hence they lack the more comprehensive and rich training content available in some of the other systems. While we have already a good coverage for both fundamental and security training, more content would help in providing more variate activities. We hope that future users of the system will begin creating new training content, and release it publicly for the benefit of everyone. We also plan to add more content ourselves, especially in the networking and security areas, for instance regarding new routing protocols.

Only IoTrain-Sim and IoTrain-Lab are open source among the compared systems, and this also makes them low cost, as students can simply use their laptops to conduct the training. On the other hand, *Smart Home Security Education* and *IoT Security Exercises* are not publicly available, and the *Watson IoT* and *3 Rocks* systems require the purchase of expensive licenses and hardware, respectively.

Regarding the multi-user aspect, IoTrain-Sim currently requires being installed by students on their own, but IoTrain-Lab makes it possible for an educator to install the system on a centrally administered server; then the system can be used in parallel by multiple users. *IoT Security Exercises* and *Watson IoT Academy* are also multi user given their virtual/online nature, but the other systems need to be physically accessed by individual users.

Finally, regarding system management, IoTrain-Sim and IoTrain-Lab were

Table 4: Performance characteristics of IoTrain-Sim

Operation	**Duration [s]**
Start Instant Contiki	26.3
Start IoTrain-Sim	0.5
Open tutorial	0.7
Open simulation	3.3
Complete simulation	8–1200
Quit IoTrain-Sim	0.3
Shut down Instant Contiki	4.9

designed to be easy to manage in terms of content and deployment. *Smart Home Security Education* and *IoT Security Exercises* are built based on custom platforms and exercise content, so adding new content is not straightforward. *Watson IoT Academy* is accessed via a web interface, so management is also straight forward, but the custom-hardware nature of the *3 Rocks Technology Systems* make content development and system setup more difficult.

We conclude that IoTrain-Sim and IoTrain-Lab meet most implementation requirements, supporting users of different levels, being open source, having a low cost for use and training content creation, and being easy to manage. IoTrain-Lab has an additional advantage due to the multi-user support we implemented. *Smart Home Security Education* fails to meet most requirements due to its closed and custom nature. *IoT Security Exercises*, on the other hand, is a strong contender that would benefit if it would be released publicly. *Watson IoT Academy* meets most of the requirements, except for the open-source and low-cost aspects, which is understandable given that it is a commercial platform. The *3 Rocks Technology Systems*, on the other hand, fail to meet most of the requirements, mainly due to their intrinsic custom-hardware nature.

6.2. Performance Evaluation

In order to evaluate the expected user experience of IoTrain-Sim and the computation load it introduces, we assessed its performance characteristics by timing basic operations, such as starting the Instant Contiki VM, running the system, opening tutorials and simulations. In this way we were able to assess the performance of IoTrain-Sim itself and of its external components in an independent manner.

The performance evaluation experiments regarding IoTrain-Sim were performed on a MacBook Air computer with a 2.2 GHz Intel Core i7 CPU and 8 GB RAM. The Instant Contiki VM used was version 3.0, which is based on the Ubuntu 14.04 Long-Term Support (LTS) operating system and Python 2.7.6. Time measurements were performed three times for each operation, and the values shown in Table 4 are averages of those measurements.

Based on the numerical results shown in Table 4, we note the following regarding the user experience and computation load of IoTrain-Sim:

- The time needed to start/shut down the Instant Contiki VM using VMware depends on computer performance, and it is not related to the IoTrain-Sim system itself. The largest duration was measured for starting the Instant Contiki VM, which at around 26 s is acceptable. Moreover, this operation only needs to be done the very first time IoTrain-Sim is used.

- Starting and quitting IoTrain-Sim, as well as opening training tutorials, are very basic tasks, and they all took less than 1 s in our experiments, showing that the system implementation is efficient.
- Opening simulations requires starting the Cooja simulator, so the duration is somewhat longer, but the time we measured was still in the order of a few seconds.
- The time needed to complete a simulation is highly dependent on its content. Thus, scenarios with more motes and a longer logical duration of the simulation will be slower to complete. For this reason, attack simulations in particular, which generate more traffic, consume a lot of computer resources, and can take up to 20 minutes for our built-in training scenarios. On the other hand, very basic simulations finished in just 8 s.

Based on the above analysis, we conclude that the user experience and computation load of IoTrain-Sim are acceptable.

6.3. User Evaluation

The systems we developed were used in our university to provide IoT training to various students. On one of these occasions we conducted a usability evaluation test by using the System Usability Scale (SUS) questionnaire [31]. Among the advantages of using SUS for system usability evaluation, we note that SUS is easy to administer to participants, and it provides reliable results even on small sample sizes; furthermore, SUS can be used to effectively differentiate between usable and unusable systems. SUS consists of a 10-item questionnaire with five response options ranging from “Strongly agree” to “Strongly disagree,” as shown in Figure 10. The questionnaire will produce a score in the range from 0 to 100, with scores larger than 68 being considered above average.

We have used the SUS questionnaire with a group of 5 students who have used IoTrain-Lab to conduct training by going through all the nine topics presented in Section 5.2, both for fundamental training with various devices and protocols, and security training via resource attacks. The detailed results are shown in Table 5, with the score for each user shown in the bottom row.

As it can be seen, the resulting SUS scores ranged from 60.0 to 85.0, with the average SUS score being 74.5. According to the information available in [32], this can be approximately converted to a percentile rank of 70% and can be interpreted as a grade of a `B-`. Therefore, we conclude that IoTrain-Lab has an overall usability that can be considered good even for learners with no cybersecurity background, as some of the responders were. Additional questioning revealed that the lower scores were caused by the fact that the experiments on FIT IoT-LAB were conducted via command-line execution, which two of the respondents were not familiar with. While the number of respondents in our

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

Figure 10: System Usability Scale (SUS) questionnaire [31].

Table 5: User evaluation results via the SUS questionnaire

SUS Item	**User 1**	**User 2**	**User 3**	**User 4**	**User 5**
1	4	5	3	4	5
2	1	2	1	3	1
3	4	4	4	4	4
4	2	3	3	3	3
5	5	5	5	4	3
6	1	1	1	2	1
7	4	4	5	3	5
8	1	4	2	2	1
9	4	3	4	3	5
10	2	4	3	4	3
Score	85.0	67.5	77.5	60.0	82.5

evaluation was relatively low, we believe we were able to capture the main aspects regarding the usability of our system. If possible, we would like to conduct a larger-scale evaluation in the future for a more thorough assessment.

6.4. Discussion

The IoT training methodology that we introduced in Section 3 is an important contribution of this paper, as it provides an outline that educators and instructors can follow when designing and implementing IoT training systems. In particular, the requirements shown in Section 3.2 can be used to guide the development so as to ensure that the resulting system applicability is as high as possible. An illustration of how these requirements can be used in practice to assess the capabilities of IoT training systems was shown in Table 3.

Another important contribution of our research is the proposal of the training content structure introduced for each of the implemented systems. This structure can be considered a "blueprint" for IoT training content development, even outside the scope of IoTrain-Sim and IoTrain-Lab. Such a thorough structure, which organizes content in clearly defined categories and sub-categories, makes it possible both for educators to create content in a comprehensive manner, and for trainees to easily locate their content of interest.

An additional content-related novel aspect of our work is that we provide both reference and attack scenarios for security training. This makes it possible for trainees to do a comparative analysis of the two in order to investigate the effects of IoT security attacks. Furthermore, the tutorials include details about how the trainees could modify the source code to perform such attacks themselves, thus letting them gain knowledge about the attack techniques. With this knowledge, trainees can then design defense mechanisms, and verify whether those mechanisms are effective, again by comparison with the reference scenarios. This deep understanding of all the issues related to IoT security will allow learners to both design and develop secure IoT systems in the future, and also to respond to the IoT security emergencies that will certainly arise at some point or another of their professional life.

While the target of our research is different, a recent survey on the use of IoT and wearable technologies in education has demonstrated the potential these technologies have in areas such as medical or vocational education and training [33]. We envisage that simulation/testbed-based approaches as the one presented here could be extended to cover other aspects of education that would benefit from the use of IoT technologies.

7. Conclusion

In this paper we presented a methodology for IoT training, including security training, and stated six requirements for IoT training system development. Then we introduced two open-source systems that we designed and implemented by following this methodology, IoTrain-Sim and IoTrain-Lab.

IoTrain-Sim employs a simulation-based approach to IoT training, making use of the Cooja network simulator to allow realistic experiments with emulated nodes using the Contiki OS. IoTrain-Sim uses both tutorials and predefined simulation scenarios to lead users from beginner to advanced level in a Learning–Viewing–Doing paradigm. This simulation-based approach reduces considerably development costs, and makes it possible to safely run a wide range of scenarios.

IoTrain-Lab takes a different approach and employs the FIT IoT-LAB testbed to make it possible to conduct experiments with real IoT devices in a multi-user approach, with no setup steps required for the trainees. Several tutorials are included to familiarize learners with the actual use of IoT devices and protocols, and a security training scenario is also provided. The tutorials are organized in increasing order of difficulty, allowing for a gradual learning experience.

We also developed an exhaustive training content structure, both for fundamental and security training, and have populated this structure with tutorial and scenarios that exercise all the features of IoT devices for fundamental training (actuation, control, sensing, communication), followed by various classes of WSN security attacks. The security content allows trainees to do forensic, attack and defense training, thus gaining full knowledge about IoT security issues.

We evaluated the two systems from a functionality perspective, and we determined that their most important advantages are learner support, availability, extensibility, flexibility and scalability. IoTrain-Sim and IoTrain-Lab also meet most of the IoT training requirements that we have formulated. While they are not yet content rich, their open-source nature makes it possible for third parties to extend the systems and develop new training content. The performance evaluation of IoTrain-Sim has shown that it is relatively lightweight, and most processing steps are performed in under 1 s. As for IoTrain-Lab, the overall usability of the platform was evaluated using the SUS questionnaire, and the resulting average score of 74.5 shows an overall good system usability.

Since we plan to use the two systems for IoT training activities in our university, one future aspect is related to developing additional training content, especially in the networking training category, such as routing, and also new security training content. An extension of IoTrain-Sim to support the new Contiki OS version, Contiki-NG, is also planned, and for IoTrain-Lab we consider extending the tutorials to use other types of nodes and OSs supported by FIT IoT-LAB, so that trainees can have an even richer practical experience.

Acknowledgment

The authors would like to thank Chunqi Du for her contribution to the development of the IoTrain-Sim GUI.

References

[1] S. Sinha, State of IoT 2021: Number of connected IoT devices growing 9% to 12.3 billion globally, cellular IoT now surpassing 2 billion, https://iot-analytics.com/number-connected-iot-devices/ (2021).

[2] H. Griffioen, C. Doerr, Examining Mirai's battle over the Internet of Things, in: Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, 2020, pp. 743–756.

[3] H. Aldowah, S. Ul Rehman, I. Umar, Security in Internet of Things: Issues, challenges and solutions, in: F. Saeed, N. Gazem, F. Mohammed, A. Busalim (Eds.), Recent Trends in Data Science and Soft Computing, Springer International Publishing, Cham, 2019, pp. 396–405.

[4] Ministry of Economy, Trade and Industry (METI), Guide to building a cybersecurity team and securing human resources, v1.1 (in Japanese), `https://www.meti.go.jp/press/2021/04/20210426002/20210426002.html` (2021).

[5] Cyber Range Organisation and Design (CROND), IoTrain-Sim GitHub page, `https://github.com/crond-jaist/iotrain-sim` (2023).

[6] Cyber Range Organisation and Design (CROND), IoTrain-Lab GitHub page, `https://github.com/crond-jaist/iotrain-lab` (2023).

[7] Coursera Inc., Cybersecurity attack and defense fundamentals specialization, `https://www.coursera.org/specializations/cybersecurity-attack-and-defense` (2023).

[8] T. Omiya, D. Fall, Y. Kadobayashi, IoT-Poly: An IoT security game practice tool for learners motivation and skills acquisition, in: Proceedings of the 19th Koli Calling International Conference on Computing Education Research, Koli Calling '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 1–10.

[9] Z. Trabelsi, IoT based smart home security education using a hands-on approach, in: 2021 IEEE Global Engineering Education Conference (EDUCON), 2021, pp. 294–301.

[10] S. Shin, Y. Seto, Development of IoT security exercise contents for cyber security exercise system, in: 2020 13th International Conference on Human System Interaction (HSI), 2020, pp. 1–6.

[11] The OWASP Fooundation, OWASP WebGoat, `https://owasp.org/www-project-webgoat/` (2023).

[12] International Business Machines Corporation, IBM Watson IoT Academy, `https://www.ibm.com/training/watsoniot/` (2023).

[13] International Business Machines Corporation, IBM Watson IoT Platform, `https://https://internetofthings.ibmcloud.com/` (2023).

[14] Cisco Networking Academy, IoT fundamentals: IoT security, `https://www.netacad.com/courses/cybersecurity/iot-security` (2023).

[15] SANS Institute, SEC556: IoT penetration testing, https://www.sans.org/cyber-security-courses/iot-penetration-testing/ (2023).

[16] 3 Rocks Technology, Internet of Things training systems, https://www.3rockstech.com/index.php/training-systems/internet-of-things (2023).

[17] Scientific & Technical Products (STP), IoT training systems, https://scitech.com.my/product-category/solution/iot/ (2023).

[18] J.-H. Cheng, H.-H. Lin, J.-H. Shen, B.-C. Chen, Z.-L. He, IoT training system for smart manufacturing education, in: IEEE International Conference on Knowledge Innovation and Invention (ICKII), 2020, pp. 182–184.

[19] Coursera, Cybersecurity and the Internet of Things, https://www.coursera.org/learn/iot-cyber-security (2023).

[20] GetSmarter, Internet of Things: Business implications and opportunities, https://mit-online.getsmarter.com/presentations/lp/mit-internet-of-things-online-short-course/ (2023).

[21] Telecoms & Tech Academy, IoT security fundamentals, https://telecomstechacademy.com/course/iot-security/ (2023).

[22] W. Shi, A. Haga, Y. Okada, Web-based 3D and 360° VR materials for IoT security education and test supporting learning analytics, Internet of Things 15 (2021) 100424.

[23] M. Heimbach, K. Holzmann, P. Stein, L. Stief, P. Berberat, M. Dirmeier, How to... train your skills goes digital! a project report on the development and implementation of practice-oriented digital student tutorials, GMS journal for medical education 39 (2022) Doc5. doi:10.3205/zma001526.

[24] Y. Yin, I. Gurvich, S. McReynolds, D. Seys, J. A. Van Mieghem, Learning by doing versus learning by viewing: An empirical study of data analyst productivity on a collaborative platform at EBay, Proceedings of the ACM on Human-Computer Interaction 2 (CSCW) (2018) 1–27.

[25] Contiki Development Team, Contiki OS GitHub page, https://github.com/contiki-os/contiki (2018).

[26] FIT IoT-LAB Development Team, FIT IoT-LAB web page, https://www.iot-lab.info/ (2023).

[27] C. Adjih, E. Baccelli, E. Fleury, G. Harter, N. Mitton, T. Noel, R. Pissard-Gibollet, F. Saint-Marcel, G. Schreiner, J. Vandaele, T. Watteyne, FIT IoT-LAB: A large scale open experimental IoT testbed, in: IEEE 2nd World Forum on Internet of Things (WF-IoT), 2015, pp. 459–464.

[28] H. Hu, Y. Han, H. Wang, M. Yao, C. Wang, Trust-aware secure routing protocol for wireless sensor networks, ETRI Journal 43 (4) (2021) 674–683.

[29] A. Mayzaud, R. Badonnel, I. Chrisment, A taxonomy of attacks in RPL-based Internet of Things, International Journal of Network Security (IJNS) 18 (3) (2016) 459–473.

[30] S. Cakir, S. Toklu, N. Yalcin, RPL attack detection and prevention in the Internet of Things networks using a GRU based deep learning, IEEE Access 8 (2020) 183678–183689.

[31] U.S. General Services Administration, System Usability Scale (SUS), `https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html` (2023).

[32] J. Sauro, Measuring usability with the System Usability Scale (SUS), `https://measuringu.com/sus/` (2023).

[33] M. Al-Emran, S. I. Malik, M. N. Al-Kabi, A survey of Internet of Things (IoT) in education: Opportunities and challenges, in: A. E. Hassanien, R. Bhatnagar, N. E. M. Khalifa, M. H. N. Taha (Eds.), Toward Social Internet of Things (SIoT): Enabling Technologies, Architectures and Applications: Emerging Technologies for Connected and Smart Social Objects, Springer International Publishing, Cham, 2020, pp. 197–209.