Title	Hands-on IoT Security Training Using IoT Testbeds
Author(s)	趙, 敏
Citation	
Issue Date	2019-09
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/16217
Rights	
Description	Supervisor: BEURAN Razvan, 先端科学技術研究科, 修士(情報科学)

Master's Thesis

Hands-on IoT Security Training Using IoT Testbeds

1710428 ZHAO, Min

Supervisor	Research Associate Professor Razvan BEURAN
Main Examiner	Research Associate Professor Razvan BEURAN
Examiners	Professor Yasuo TAN
	Associate Professor Yuto LIM
	Research Associate Professor Ken-ichi CHINEN

Graduate School of Advanced Science and Technology
Japan Advanced Institute of Science and Technology
(Information Science)

August 2019

Abstract

Nowadays, the IoT (Internet of Things) is not a virtual concept anymore, it is becoming a quite usual thing which can be seen everywhere in our lives and slowly filtering into people's minds.

From smart home to smart cars, smart industries, smart cities, etc., this new situation is quickly arising that sensors are connected to the Internet, collect data actively or receive the instructions passively, and transmit data, instructions to the cloud for processing, and then provide plentiful of services for people or companies. Although the concept that "machines will assist human in daily life" had already been described in 1932, in 2020, people will be surrounded by 50 billion Internet connected devices based on predictions of Cisco.

However, while people enjoy these conveniences via technology, an inevitable problem is the security of IoT. With the rapid development of IoT, cyber-attacks occur worldwide day by day. A notorious conducted DDoS attack case happened in 2016, named Dyn cyberattack, which was using a large number of IoT devices, such as printers, IP cameras that had been infected with the Mirai malware. As a matter of fact, Gartner also predicts that from 2018 to 2021, the worldwide spending on IoT security will increase from $1.5 billion to $3.1 billion. Thus, the focus on IoT security issues is becoming more and more critical.

Comparing with theoretical lectures, we believe that, under a complex and volatile security environment, it is necessary to do specific hands-on training, which can give trainees not only the necessary background information, but also indispensable, precious practice experience, in order to handle the rapidly and continuously changing security issues in an efficient way. Besides, we found that preparing a training environment is necessary for some training tutorials, but it is also like a stumbling block for beginners. Thus, it is useful to provide an online, "access-and-use" style, user-friendly training environment. Moreover, for the instructors, this training environment needs to be managed lightly and efficiently.

This problem does not exist for trainees like students only. As a matter of fact, in practice, only a small percentage of professionals are exposed to the production environments directly, and they do not make decisions. People who actually make security, technical or strategic decisions have little idea about the front-line environment. In this situation, during the promotion of emerging technology, should we just wait at ease, embrace it without borders, or use it for our benefit in a targeted way? The answer is obviously the last

one. But how to use technology in a targeted way? One of the solutions is to test and implement it in a virtual training environment first. Thus, questions like how to identify the standards of best practices and how to adjust measures to specific conditions will smoothly be solved.

This thesis presents IoTrain-Lab (IoT Training Platform using FIT/IoT-LAB Testbed), which is an open-source, easy to maintain, migrate and expand platform that also supports for multiple users' training for those who would like to know about IoT or IoT security. Furthermore, it contains both fundamental training and security training for meeting different demands.

There is no doubt that, for practical and flexibility reasons for designing, implementing, and testing experiments, simulators, emulators, and testbeds are beneficial tools. However, we have to consider the gap between simulation and reality. As we know, the simulator can change parameters as instructor wants, the ideally simulated environment may lead to single result set compared to real deployments. A satisfactory solution for these issues is using physical testbed. We did a survey and compared the differences between various IoT testbeds, and an open-source, open-access, multi-user testbed comprised of several wireless nodes and mobile robots, named FIT/IoT-LAB, became main candidate.

FIT/IoT-LAB is part of OneLab (Internet-overlaid, Broadband access, wireless & loT), and it has six sites across France. Each site has different devices, but all of these sites are related, and users can use the same web portal to access the devices. Moreover, FIT/IoT-LAB provides three kinds of hardware, which represent low complexity devices, middle devices, and rich devices. Except for these static nodes, another type is mobile nodes, robots which can be used at all sites. These up-to-date, programmable devices are free to use for research.

From the instructor point of view, considering the time cost issues, IoTrain-Lab was designed and implemented on Docker, an emerging virtualization approach solution. The instructor can implement IoTrain-Lab through the Docker files and Docker-compose files, which is efficient, and easier for maintenance and expansion. Comparing with the traditional virtualization way of virtual machines, Docker has several advantages, such as a more efficient use of system resources, faster start-up time, easier migration, easier maintenance and expansion, etc. When an instructor creates tutorials based on different demands, they can release it to Moodle web page, at the same time, trainees can use their own browser to access the Moodle web page and get the training tutorials. We also provide a Linux training environment which is connected with a clientless remote desktop gateway and it is also an online and "access-and-use" environment.

There are three clear classifications for fundamental training from top to

down, Application, Network Protocols, and Devices. This part is mainly for trainees who do not have much experience in the IoT area. It introduces the experiments from basic to complex, from a single node to several nodes, from perception, communication to application. For example, in the Devices part, we divided it into two subparts, consumption and radio in order to match the different training targets. For the consumption, we gave an experiment about battery consumption monitoring, and the trainees can do the hands-on practice based on this tutorial and FIT/IoT LAB Testbed, so that trainees can get results and analyze the results based on the testbed.

For trainees who already have an understanding of IoT, or would like to extend their knowledge to IoT security area, a similar structure with fundamental training includes IoT Service Ecosystem, Communication Network Ecosystem and Endpoint Ecosystem from top to down. This structure is based on the GSMA IoT security model, which promotes the best practice for secure design, development, and deployment of IoT services. These interrelated training contents ensure trainees have as many choices as they need.

In this thesis, first, we describe the research background, motivation and a short definition of the Internet of Things and current situation for IoT security. Then we introduce what an IoT testbed is, what is the difference between FIT/IoT-LAB and other testbeds, the architecture of IoTrain-Lab, furthermore, the necessary external tools like Docker and Apache Guacamole. Next, we address the training content and we offer precise training content details for both fundamental training and security training based on individual structures, however, due to various conditions, only the details and implementation of Flooding attack is provided and explained in this paper, and other parts are still on-going. At chapter five, two evaluation methods will be given, feature evaluation and user evaluation via SUS (System Usability Scale) which is a reliable tool for measuring usability. Last, conclusion and future work will be presented.

Keywords: IoT security training, IoT Testbed.

Acknowledgements

As the saying goes, time flies when you're having fun. I am extremely cheerful that I can have an unforgettable memory in JAIST, both in research and everyday life. Wherever am I, I won't forget it.

First and foremost, I would like to express my special thanks of gratitude to my supervisor professor Razvan Beuran, who provided me many precious opportunities to grow up. During these two years, there were many things I learned from him, not only academically but also in life. He gave me a lot of significant suggestions and guidance which let me have more courage and confidence complete the master program, without his generous help this thesis would not have been finished smoothly.

Second, I want to thank my friends who help me all the time, thanks for all the supporting, caring, raising up which without paying from them. We knew each other from the same for music habits. I am so lucky that I can meet such friends who shares my ambitions and outlook on life. Thanks for all these lovely guys.

Last but not least, I would like to express my sincere gratitude to my family, especially my father, who does not speak English or Japanese, has no idea about computer or cyber-security at all, but he tried his best to support me, he always encourages me to do what I want to do, gives me the courage to face a difficult time. Without his financial supporting, I will not have a chance to study abroad.

Thanks to all the people who I met on the way, thanks to all the experiences which made me belly laughing or crying, all of these are adored remembrance for me.

Contents

1 Introduction 1

2 Research Background 3

2.1 Background . . . 3

2.2 Motivation . . . 5

2.3 Internet of Things . . . 6

2.3.1 Definition . . . 6

2.3.2 Architectures . . . 8

2.4 The Security of IoT . . . 9

3 Training Platform Using IoT Testbeds 11

3.1 IoT Testbeds . . . 11

3.2 IoT Testbed Comparisons . . . 13

3.3 FIT/IoT-LAB Testbed Details . . . 16

3.4 IoTrain-Lab . . . 19

3.5 Extend tools . . . 23

3.5.1 Linux container . . . 23

3.5.2 Docker . . . 25

3.5.3 Apache Guacamole . . . 25

4 Training Content 27

4.1 Content Overview . . . 27

4.2 Fundamental Training . . . 29

4.2.1 Application . . . 29

4.2.2 Network Protocols . . . 29

4.2.3 Devices . . . 31

4.3 Security Training . . . 32

4.3.1 Endpoint Ecosystem . . . 32

4.3.2 Communication Network Ecosystem . . . 33

4.3.3 IoT Service Ecosystem . . . 35

5 Evaluation **36**
5.1 Feature Evaluation 36
5.2 User Evaluation 39

6 Conclusion **42**

References **44**

List of Figures

2.1 Top 10 IoT Segments in 2018. 3
2.2 Japan IoT market. 4
2.3 The imagination of IoT. 6
2.4 Technological and social aspects. 7
2.5 IoT three layers. 8
2.6 OWASP IoT Top 10 vulnerabilities in 2018. 10

3.1 The proportion of simulation and experiment. 12
3.2 How to use FIT/IoT-LAB. 13
3.3 Classification of testbeds. 14
3.4 Various hardware platforms in FIT/IoT-LAB. 16
3.5 The mobile nodes in FIT/IoT-LAB. 17
3.6 Platform architecture. 19
3.7 Using docker-compose.yml and Dockerfile to create CentOS. . 20
3.8 The running containers. 21
3.9 The training environment. 21
3.10 Contents creation process. 22
3.11 The difference between containers and virtual machines [1]. . . 24
3.12 The comparing of docker containers and virtual machines [2]. . 25
3.13 The architecture of Apache Guacamole [3]. 26

4.1 GSMA IoT model [4]. 27
4.2 Training content overview. 28
4.3 Application structures. 29
4.4 Network protocols structures. 30
4.5 Devices structures. 31
4.6 Communication Network Ecosystem structures. 33
4.7 Flooding Attack. 34
4.8 The impact of Flooding Attack for UDP server and client. . . 34

5.1 System configuration of Internet of Things Trainer. 37
5.2 System usability scale [5]. 39

5.3 A comparison of questionnaires [6]. 40
5.4 SUS Score [7]. 41

List of Tables

3.1 Comparison of open IoT Testbed. 15
3.2 Operating systems availability [8]. 17

5.1 Feature comparison 38
5.2 SUS Calculation 41

Chapter 1

Introduction

With the advent of emerging technologies, many things that people could not imagine before have gradually turned from concept to reality, gradually entering people's lives, such as smart home to smart cars, smart industries, smart cities, etc. Although the phrase "Internet of Things" first coined [9] in 1999, according to a Cisco prediction [10], from 2015 to 2020 the number of devices connected to the Internet will increase from 25 to 50 billion. In 2020, the connected devices per person will increase to 6.58. Thus, it is reasonable to believe that IoT will become more and more closely bound with people's life.

However, when people have been immersed in the convenience after smart things, one issue that cannot be ignored is security. In 2016, there was a DDoS attack that targeted the DNS provider Dyn, which affected Europe and North America, especially the Eastern United States. The method used in this attack is Mirai botnet. There is no doubt that the more devices we have, the more risks from the vulnerabilities of these devices we should be concerned with. According to a Gartner analysis prediction [11] from 2018 to 2021 the worldwide spending on IoT security will increase from $1.5 billion to $3.1 billion. Based on this complex and volatile security environment, we believe that doing hands-on security training and education is the only way to prevent, discover, handle such security incidents.

In addition, we found that many training cases required trainees to prepare a training environment or complete the configurations which is error-prone and easy to be frustrated like a stumbling block for beginners.

Thus, we decided to design and implement an online, user-friendly training platform, including both fundamental training and security training, named IoTrain-Lab. The instructor creates training content and input it to a Moodle container, then, trainees open their own browser access and the Moodle webpage to get the training content.

We anticipate the advantages of this research to be:

(i) It can attract people's attention to IoT security, not only for the practitioners like security professionals, researchers, or engineers, but also for other people who have interest in the IoT area.

(ii) For people who are working in the IoT area, not only the IoT training structure can be enhanced, but it can also inspire various new ideas.

(iii) For trainees who did this training, they will be able to grasp quite complex knowledge. In the meantime, theoretical learning and hands-on practice also can inspire trainees' creativity.

(iv) Considering the time/cost issue, IoTrain-Lab should also be easy to manage for instructors as well.

(v) This research can also contribute a meager strength to the IoT security training industry as IoT security plays an important role in today-of-art technology. In the meantime, other researchers who hold similarly interest points with us might receive a few references and inspiration.

Already in 1932, Jay B. Nash described that machines will assist humans in daily life, how our lives will be when inter-connectivity of devices takes center stage. Nowadays, billions of machines connected with the network became smart and humanize that gives people warm butler and humanized service. Standing on the shoulders of the blazers in the IoT area, we consider that the contributions of this research are as follows:

- We designed both architecture for IoTrain-Lab and training contents which includes fundamental training and security training.

- We developed the platform on Ubuntu OS, it is lightweight and easy to be managed due to all the tools were implemented based on Docker and we provided the training contents via Moodle web page, which is highly realistic due to taking place on an actual testbed.

- We evaluated the IoTrain-Lab from feature evaluation and user evaluation, comparing with others, our platform is simple, fast and user-friendly for both instructors and trainees.

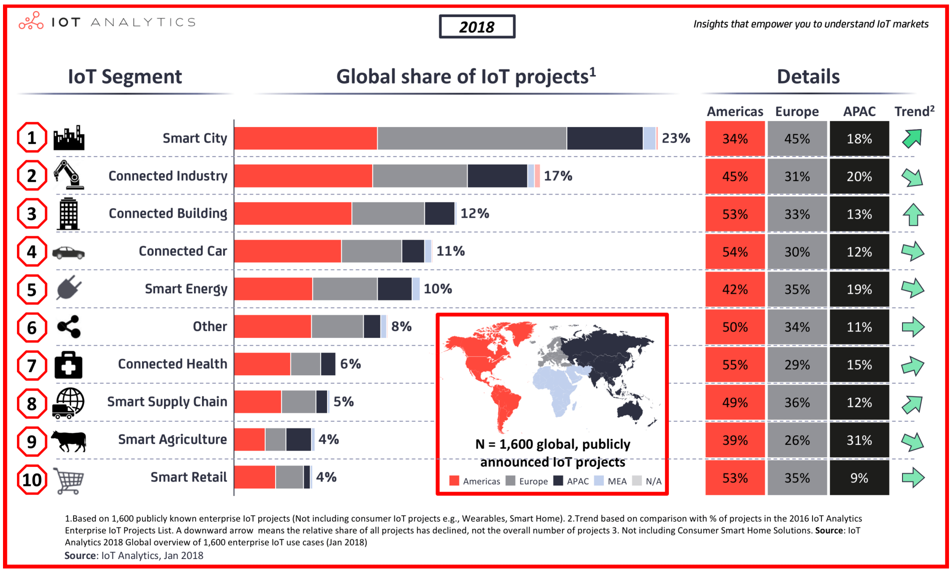
Chapter 2

Research Background

For this chapter, we give a necessary research background in section one, such as what the current IoT market does look like? what kind of IoT projects do people focus on? For section two, we present the motivation for doing this project. Section three introduces a brief concept of IoT. Last, we present the IoT security issues we meet for now.

2.1 Background

Currently, the size of the IoT market is growing year by year, According to the source from IoT analytics [12], we got the details in Figure 2.1.



1.Based on 1,600 publicly known enterprise IoT projects (Not including consumer IoT projects e.g., Wearables, Smart Home). 2.Trend based on comparison with % of projects in the 2016 IoT Analytics Enterprise IoT Projects List. A downward arrow means the relative share of all projects has declined, not the overall number of projects 3. Not including Consumer Smart Home Solutions. **Source**: IoT Analytics 2018 Global overview of 1,600 enterprise IoT use cases (Jan 2018)

Source: IoT Analytics, Jan 2018

Figure 2.1: Top 10 IoT Segments in 2018.

We can find that the top three IoT projects in progress are Smart City, Connected Industry and Connected Building, the proportion of this progress is 23%, 17% and 12%. More specifically to say, Europe more cares about Smart City, 45% Smart City projects are located in Europe, for Americas, they are strong in Connected Health(55%), Connected Car(54%), Smart Retail(53%) and Connected Building(53%). In Japan, IoT is also a hot and popular industry, according to the report from Universal Data Resources Inc [13], we can find the details in Figure 2.2 that from 2016 to 2022, the amount of user spending in Japan's IoT market will increase from 4.8 trillion yen to 12.3 trillion yen.

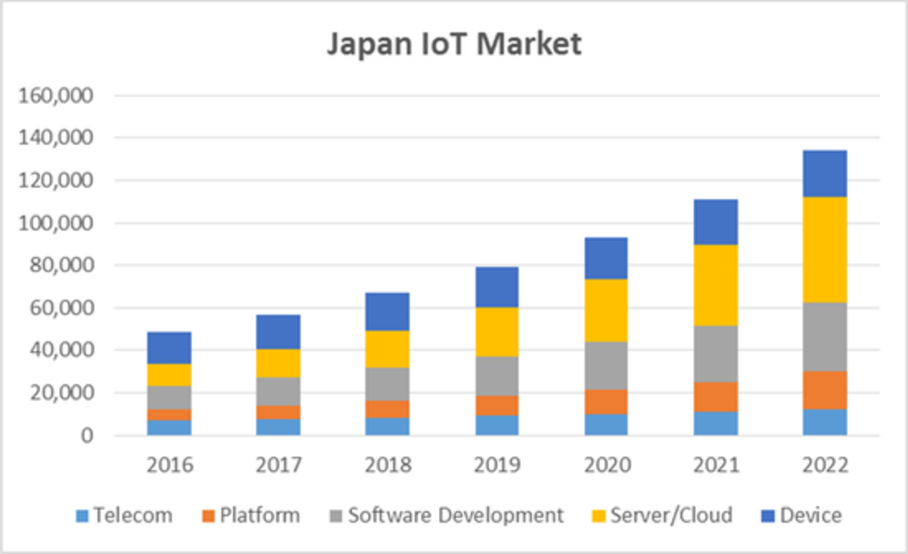


Figure 2.2: Japan IoT market.

For the spending in the IoT market in Japan, IT research company IDC Japan pointed that [14], about the market size of IoT in Japan, the amount spent in 2017 is 5.816 trillion yen, they predicted that the annual average growth rate (CAGR: Compound Annual Growth Rate) within the forecast period grows at 15.0%, and the amount spent in 2022 is It is expected to be 11.7 trillion yen.

With the trend of business-driven technological advances and technology-led business development, it has been a research field of particular significance to study and control the security risks brought by new technology, such as IoT security. As we get the information from newspapers, magazines, or news sites, the IoT security issues were never stopped. In 2016, a Dyn cyberattack became famous because it used a large number of IoT devices such as printers, IP cameras, etc.. In 2017, the attacks against IoT devices were still up by

600 percent last year, it is too common to find that default passwords and unpatched vulnerabilities are still the worst-hit area [15]. In 2018, from the report provided by Cisco [16], they clearly pointed that, with the development of IoT market, IoT and DDoS attacks are growing diversity, such as security cameras and tablets, to gain access to public networks.

The IT research company IDC Japan also presented that [17], comparing with the previous year, the size of Japan IoT security product market in 2017 was increased by 20.5% reached 62.4 billion yen, and the average annual growth rate (CAGR: Compound Annual Growth Rate) in 2017-2022 is 14.4%, in 2022, is expected to expand to 122.1 billion yen, twice as much as 2017.

All of these emergency situations can reflect a phenomenon that we need more attention for IoT and the security for IoT, however, there is a huge gap between book knowledge and actual work because of the speed of the devices update at school course, the time of preparing hands-on practice environments, etc., which also means we still have abounding potentials and opportunities in IoT area especially for IoT security.

2.2 Motivation

There is an old Chinese saying "The water that bears the boat is the same that swallows it up [18]", security is the same. Under a growing technology environment, it goes without saying the importance of security. We believe that only give people practice hands-on training can handle these issues effectively. Thus, how to classify the training people and what kind of training contents are the points.

In [19], there is a concept called the democratization of cyber-security training, not only the practitioners like current security professionals, researchers, engineers, should attend to security, but also young people. Motivated to improve the current IoT security training methods, also the concept of democratization training. We designed training contents which has both fundamental training and security training for different demands, such as trainees who do not have much knowledge for IoT, they can start from fundamental training which can lower the threshold for learning new things, for those who have considerable knowledge for IoT or want to involved in this security field, they can start from security training. We aimed to design open-source, user-friendly, and lightweight training platform contains various training content. Our training platform can not only help trainees grasp difficult knowledge, but also can inspire their creativity through hands-on practices.

2.3 Internet of Things

There are two subsections in this section, in the first subsection, we present a brief introduction for IoT, like what kind of technological and areas does IoT cover. In the second subsection, we address the three layers termed as perception, network, and application layers.

2.3.1 Definition

Generally speaking, the IoT covers many areas like academic, industry, medical, also include system architecture, software architecture, and services, (see the imagination of IoT at Figure 2.3).

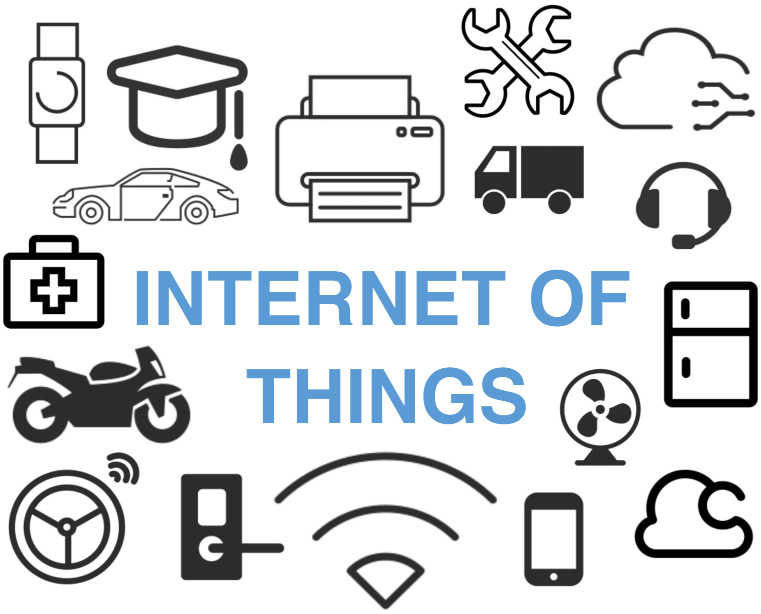


Figure 2.3: The imagination of IoT.

Normally, different perspectives and business interests will lead to different definitions and models. Nonetheless, there is one propose that, no matter what model is designed, it should show the advantages of IoT and strength its weaknesses. Due to the importance of user's privacy and protect user's personal data. Moreover, the issue of security does not merely refer to security technology; instead, it should be considered in an organizational environment. Security can support business development. Security technology should also offer support for services, in the same way, that traditional IT development and other technical support services. Thus, we considered that security and privacy should be considered at the beginning. In this thesis, we found that the IEEE IoT initiative has released a picture at [20] about what kind of areas does IoT cover, which we think is comprehensive and has much reference value, (see the details at Figure 2.4).

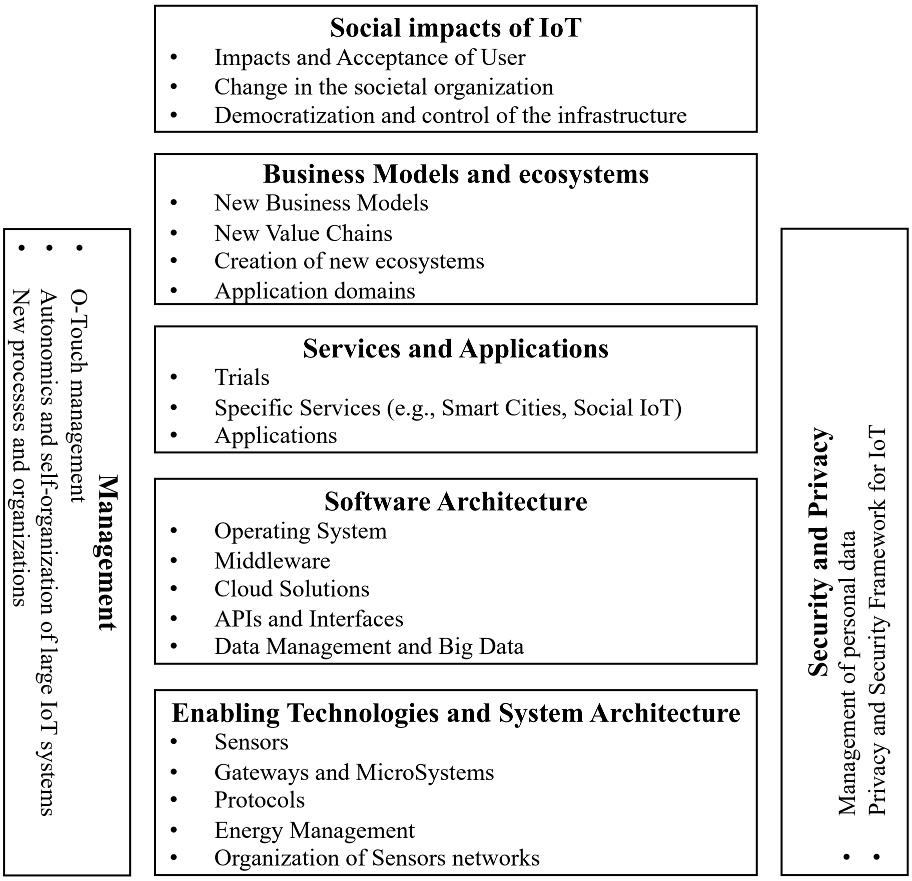

Social impacts of IoT
Impacts and Acceptance of User
Change in the societal organization
Democratization and control of the infrastructure
Business Models and ecosystems
New Business Models
New Value Chains
Creation of new ecosystems
Application domains
Services and Applications
Trials
Specific Services (e.g., Smart Cities, Social IoT)
Applications
Software Architecture
Operating System
Middleware
Cloud Solutions
APIs and Interfaces
Data Management and Big Data
Enabling Technologies and System Architecture
Sensors
Gateways and MicroSystems
Protocols
Energy Management
Organization of Sensors networks
Management
O-Touch management
Autonomics and self-organization of large IoT systems
New processes and organizations
Security and Privacy
Management of personal data
Privacy and Security Framework for IoT

Figure 2.4: Technological and social aspects.

As Figure 2.4 presented, it includes basic enabling technologies and system architecture like sensors, gateways, and micro-systems, it also involves higher components such as social impacts.

We believe that, currently, this clear definition addresses most IoT features, it can help researchers, developers, and professionals a better understanding for IoT, and further promote the development of the Internet of Things industry.

2.3.2 Architectures

Different architectures usually match different demands, typically, more and more researchers [21] [22] [23] divided the architecture into three basic layers: from top to down they are Application layer like smart home, smart city or other smart things. The network layer, which includes network protocols, WiFi, and a layer has a sensor to approach things called Perception layer. Each layer has its vulnerabilities, Figure 2.5 shows the three layers.

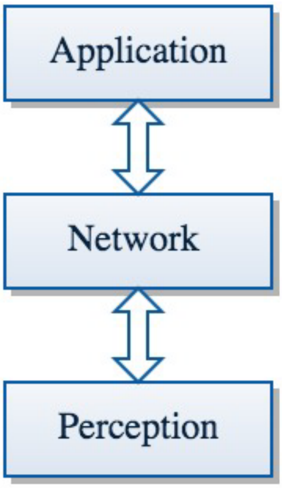


Figure 2.5: IoT three layers.

- Perception layer

For easier understanding, we also use “sensor layer” to represent the perception layer, which is a basic, important and necessary layer. Due to the perception layer can let the things “speak and publish information”, it is an important part of the integration of physical world and information world, also the unique feature of the Internet of Things that is different from other networks. Ideally, sensors, actuators at this layer catch the data from the environment and transmit it to the network layer.

- Network layer

When the perception layer received information, it should be via the network layer to send it out that user can use it. The network is one of the most important infrastructures of IoT has a link function. It is responsible for transmitting the perceptual information to the upper layer, transmitting the command to the lower layer; in short, it is transmitting the data.

- Application layer

Data from the perception layer will eventually become a convenient tool for users. This is why application layer exists, it can be regarded as a higher layer in the architecture. Basically, it represents the advanced smart environment like the smart home.

Nowadays, there are also many industries, such as medical, energy, construction, agriculture, and manufacturing, that have begun to explore more solutions in the field of IoT. Traditional Internet had been changed from data as a centering to people as a centering. Typical applications include email, online gaming, and social networking. But IoT is centered on the “thing” and the physical world.

2.4 The Security of IoT

As we presented at chapter 1 that the influence of IoT is increasing day-by-day, and the IoT security issues as well. Just a while ago, a security company Avast announced that [24]:

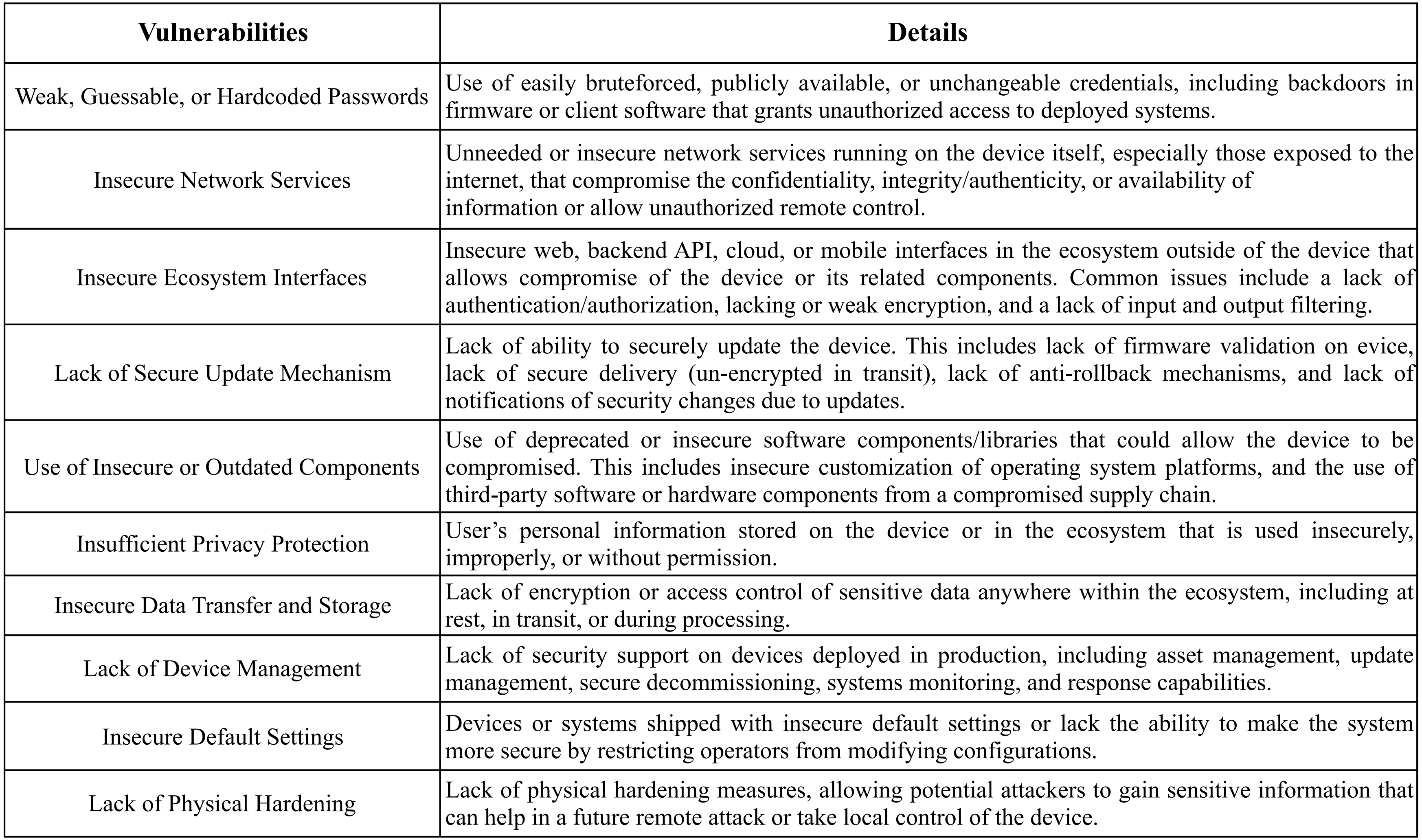
- 29% of Japanese households have at least one vulnerable device, and the entire home network is at risk (World Average is 41%)

- 57% of home routers in Japan are vulnerable (World average is 60%)

- Media streaming terminals, security cameras, and printers are extremely vulnerable except routers and network devices.

It is not just a piece of news. It is facts that are considered to be a very familiar event around us. IoT is morphing so quickly, from considerable standardization, white papers, national initiatives, etc.. We found that OWASP(Open Web Application Security Project) released IoT Top 10 vulnerabilities in 2018 [25], (see the details in Figure 2.6).

Developers, manufacturers, businesses, and consumers use it as a reference in order to avoid these security issues and improve the security of IoT. At meanwhile, we also considered that, based on this Top 10 vulnerabilities, we could design and implement a few targeted security training to help trainees master the technology that keeps pace with the times.



Vulnerabilities	Details
Weak, Guessable, or Hardcoded Passwords	Use of easily bruteforced, publicly available, or unchangeable credentials, including backdoors in firmware or client software that grants unauthorized access to deployed systems.
Insecure Network Services	Unneeded or insecure network services running on the device itself, especially those exposed to the internet, that compromise the confidentiality, integrity/authenticity, or availability of information or allow unauthorized remote control.
Insecure Ecosystem Interfaces	Insecure web, backend API, cloud, or mobile interfaces in the ecosystem outside of the device that allows compromise of the device or its related components. Common issues include a lack of authentication/authorization, lacking or weak encryption, and a lack of input and output filtering.
Lack of Secure Update Mechanism	Lack of ability to securely update the device. This includes lack of firmware validation on evice, lack of secure delivery (un-encrypted in transit), lack of anti-rollback mechanisms, and lack of notifications of security changes due to updates.
Use of Insecure or Outdated Components	Use of deprecated or insecure software components/libraries that could allow the device to be compromised. This includes insecure customization of operating system platforms, and the use of third-party software or hardware components from a compromised supply chain.
Insufficient Privacy Protection	User’s personal information stored on the device or in the ecosystem that is used insecurely, improperly, or without permission.
Insecure Data Transfer and Storage	Lack of encryption or access control of sensitive data anywhere within the ecosystem, including at rest, in transit, or during processing.
Lack of Device Management	Lack of security support on devices deployed in production, including asset management, update management, secure decommissioning, systems monitoring, and response capabilities.
Insecure Default Settings	Devices or systems shipped with insecure default settings or lack the ability to make the system more secure by restricting operators from modifying configurations.
Lack of Physical Hardening	Lack of physical hardening measures, allowing potential attackers to gain sensitive information that can help in a future remote attack or take local control of the device.

Figure 2.6: OWASP IoT Top 10 vulnerabilities in 2018.

Chapter 3

Training Platform Using IoT Testbeds

This chapter introduces the IoT Training platform using IoT testbed named IoTrain-Lab, which is an open-source, lightweight, support for multiple users' training platform. The first section answers questions like what is an IoT Testbed? Why do we use it? What are its advantages? The second section compares the difference between FIT/IoT-LAB testbed and other testbeds. The third section introduces what is FIT/IoT-LAB Testbed look like. The fourth and fifth section presents the platform architecture and implementation, which is one of the contributions of this research. The last section, a short discussion, will be presented.

3.1 IoT Testbeds

A testbed is a platform to test algorithms and protocols in order to evaluate researchers' contributions through running real experiments. For IoT testbed, it should have a number of nodes (devices) which can be used for the researcher to monitor experiments like nodes energy consumption or network topology. It is an invaluable tool that can help people who want to test, validate their solution before real implementation.

In the past decade, simulation is widely used because of the complexity and difficulty to design, implement a real testbed for experiments. There is no doubt that simulation has some advantages over testbeds, such as user can conveniently construct, modify a scenario, and collect the data, implement a huge topology without paying. Furthermore, testbed usually is infected by the environment, in [26], the author also pointed that, "a slight change in temperature or humidity can impact hardware calibration, and the closing of a door can inadvertently change the propagation of the wireless channel. "Nevertheless, the simulators have their own disadvantages, due to a user can change the parameter as they wanted, there is a gap between trustworthy real results and simulator results. In [27], the authors get this trend (Figure 3.1) through compiling 596 ad hoc and WSN related articles.

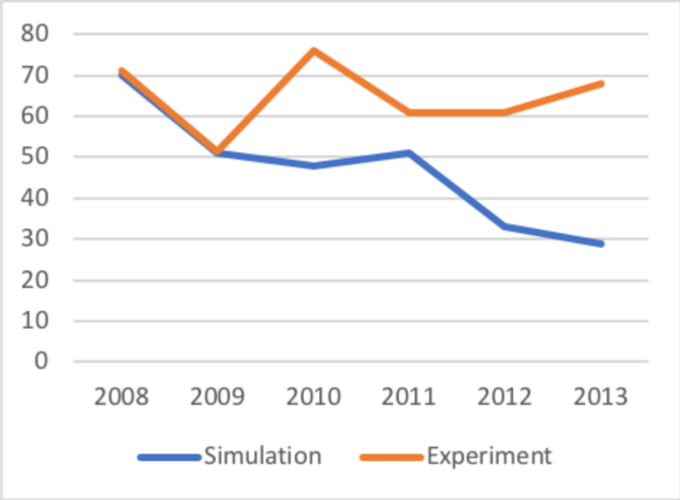


Figure 3.1: The proportion of simulation and experiment.

Though this picture, we can clearly find that, form 2008 to 2009, the number of simulation and experiment were both decreased, but from 2009 to 2013 the proportion of experiment has increased relatively stable year by year. The authors also pointed out that [28], the main reason to explain this situation could be the cost of creating testbed is decreasing. Meanwhile, they also found that over two-thirds of 596 used experiments to verify their concept. No doubt using testbed for implementation is becoming a popular trend gradually.

We think it is reasonable that with the development of new technology, the reason why more and more researchers use to experiment in their project is not only because the cost to testbed set up lower than before, but also in recent years there have been many open source testbeds that have contributed to this phenomenon like FIT/IoT-LAB Testbed. At before, people might consider the labor cost and economic cost to implement a testbed. However, the open testbeds have been developed emergency that researchers have access to use these to build a block in their projects.

In this research, we focus on using real experiments to do hands-on training, which can help trainees learn the knowledge from practice, also can get the spirit of real experimentation.

3.2 IoT Testbed Comparisons

In chapter 3.1, we described the current status for simulation and experiments, when simulation can not satisfy our demand, we should choose the experiment as an infrastructure. we believe that actual testbed is based on intuitive numbers and scientific supporting and highly realistic. Thus, we did a survey for IoT testbed for knowing the details more specifically. In this thesis, we chose five representative testbeds: FIT/IoT-LAB [8], Fed4FIRE+ [29], GENI [30], WISEBED [31] and SmartSantander [32].

FIT/IoT-LAB [8] is part of the FIT (Future Internet of the Things) platform, which belongs to the Onelab facility [33]. FIT/IoT-LAB is a large open-source testbed which can support researchers run their experiments like new protocols, solutions, or a node energy consumption, radio sniffing. It has over 1700 wireless sensor nodes located in different sites across France that can be fully controlled by users, and three kinds of main nodes, which represent low-power devices like radio chip sensors, today' s state-of-the-art IoT devices and advanced devices such as set-top boxes. Figure 3.2 represents the procedure of use FIT/IoT-LAB.

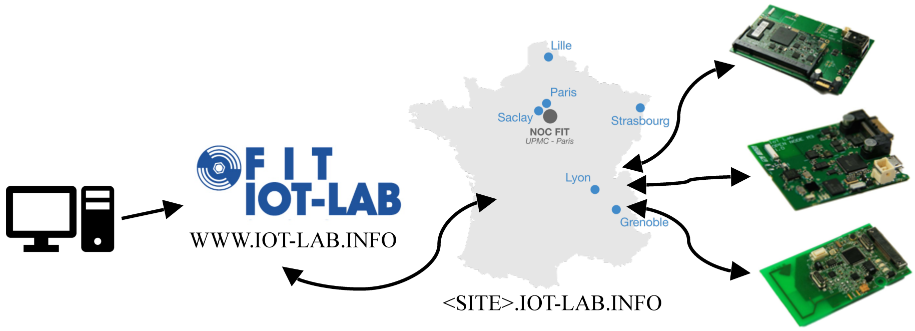


Figure 3.2: How to use FIT/IoT-LAB.

In general, there are just three steps for user use the sensors: (i) Access with FIT/IoT-LAB web-portal, user can submit experiments, flash, read and write serial port. (ii) There are fifth sites provided by FIT/IoT-LAB, and each site has different hardware. Based on different demands, that user can choose one of the sites to use. (iii) Through the site which user-chosen, they can access and control the node via UID of each node.

The FIT/IoT-LAB is an open-source, open-access, multi-user, state-of-the-art testbed comprised of wireless nodes and mobile robots; more information presents at section 3.3.

Fed4FIRE+ is a project under the European Union's Program Horizon 2020 [29]; they put seventeenth testbeds together as a federation, use common tools for interacting with testbeds. Same with FIT/IoT-Lab, they also allow the user to get open access with those testbeds to run their experiments. The area covers all the new topics like 5G, cloud computing, grid computing, IoT, and big data. We found the specific classification for testbeds in Fed4FIRE+ [29] (Figure 3.3).

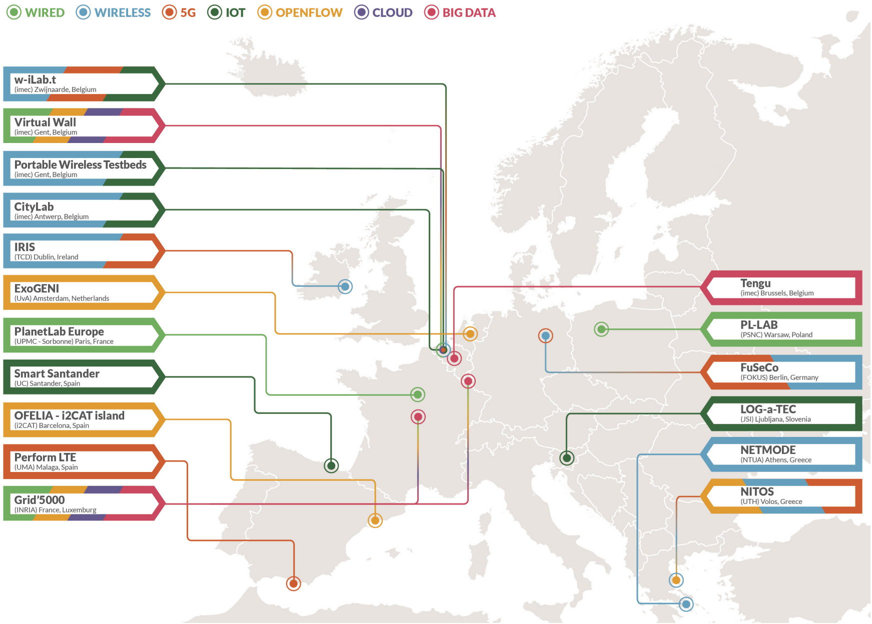


Figure 3.3: Classification of testbeds.

GENI (Global Environment for Network Innovations) provides a virtual laboratory for networking and distributed systems research and education [30], different with FIT/IoT-LAB, GENI is mainly focused on network area, for example, user can test their experiments by using Layer 2 networks in topologies via testbeds. The map of GENI is omitted because of space, see the details at their website[34].

WISEBED is a large-scale wireless sensor network testbed [31], which organized by European universities and research institutes, the propose is providing an environment for researchers, scientists, etc.. to test their sensor network-related experiments. User can get real-time data from the physical world via WISEBED.

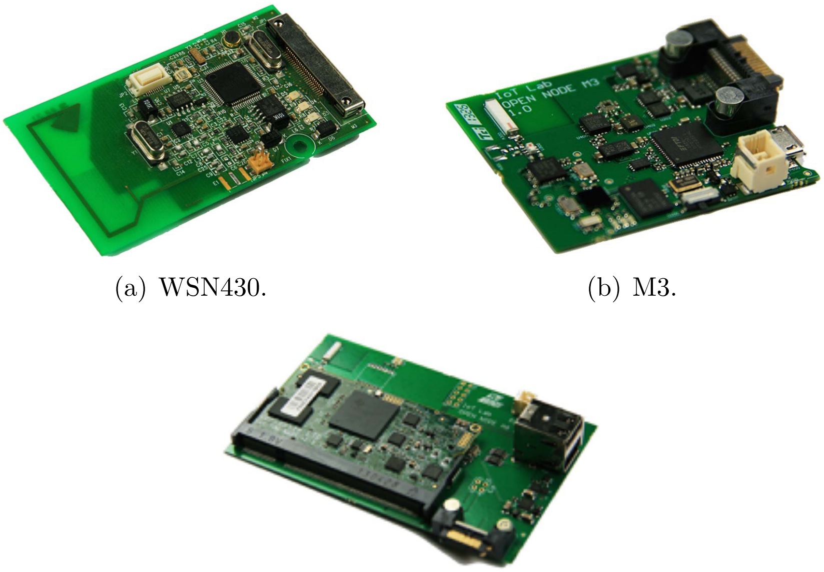
SmartSantander [32] is a small city with nearly 20,000 sensors in static objects such as parking lots, buildings, and dynamic objects like buses, taxis, and garbage trucks. All of these sensors can collect information and transmit it to the management server for analysis and prediction. Basically, it upon WISEBED and extend the domain to the outdoor area.

Testbed	Short Description	Devices	Devices Details	Mobility
FIT/IoT-LAB	A large scale open experimental IoT testbed [8]	1786 wireless sensor nodes	WSN430 Node, based on MSP430F1611 MCU and communication with an 802.15.4 PHY Layer (800 MHz or 2.4 GHz) M3 Node, based on STM32F103REY MCU and communication with an 802.15.4 PHY Layer (2.4 GHz) A8 Node, based on TI SITARA AM3505 (ARM Cortex A8) allows running Linux. This node also embeds an M3 Node with 802.15.4 comm [35]	Yes
Fed4FIRE+	The largest federation worldwide of Next Generation Internet [29]	Organized by 17 testbeds	1. CityLab 2. ExoGENI 3. FuSeCo 4. GRID'5000 5. OFELIA I2CAT island 6. IRIS 7. LOG-a-TEC 8. NETMODE 9. NITOS 10. Perform LTE 11. PL-LAB 12. Planetlab Europe 13. Protable Wireless Testbeds 14. SmartSantander 15. Tengu 16. Virtual Wall 17. w-iLAB.t [29]	Unknown
GENI	A virtual laboratory for networking and distributed systems research and education [30]	Organized by 68 testbeds, 43 of them are up current 2019/7/1	Details at https://portal.geni.net/amstatus.php	No
WISEBED	A multi-level infrastructure of interconnected testbeds of largescale wireless sensor networks [31]	550 nodes in 2009, organized by 4 testbeds	1. Trio testbed, one of the largest wireless sensor testbeds, indoor and outdoor 2. MoteLab testbed, an indoor sensor network testbed 3. TWIST testbed, resides indoor 4. TutorNet testbed, 3-tire network topology with testbed server, gateway stations, and sensor nodes [31]	No
SmartSantander	A unique in the world city-scale experimental research facility in support of typical applications and services for a smart city [32]	Around 2000 IEEE 802.15.4 devices deployed in a 3-tiered architecture	IoT node is responsible for sensing the parameters like temperature, CO, light, etc. Repeaters placed high above the ground in street lights, semaphores, information panels, etc. Gateway node collects the measurements and uploads to servers [32]	Yes

Table 3.1: Comparison of open IoT Testbed.

3.3 FIT/IoT-LAB Testbed Details

As we described in chapter 3.1, from 2009 to 2013, the number of using experiments is growing year by year. We did a comparisons of IoT testbeds at chapter 3.2 for knowing the details more specifically, we need a testbed which is open-source, various kind of sensors included, user-friendly, supporting SSH access to a testbed serve, etc.. At this time, FIT/IoT-LAB became the top candidate. Figure 3.4 and Figure 3.5 presents five main hardware in the FIT/IoT-LAB.



(a) WSN430.

(b) M3.

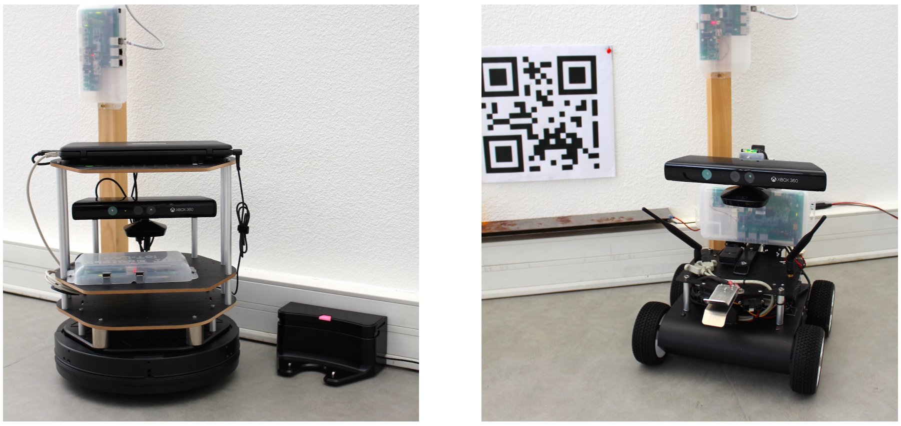
(c) A8.

Figure 3.4: Various hardware platforms in FIT/IoT-LAB.

Figure 3.4 has three sub figures, Figure 3.4(a) is a WSN430 node, based on a low power MSP430-based platform, with a fully functional ISM radio interface and a set of standard sensors [35]. There are two kind of WSN430 node in FIT-IoT/LAB, WSN430 v1.3b and WSN430 v1.4 with similar performance, both have temperature sensor and ambient sensor light, supporting various operating system, FreeRTOS [36], Contiki [37], Riot [38], TinyOS [39], OpenWSN [40].

Figure 3.4(b) is a M3 node, based on a STM32 (ARM Cortex M3) microcontroller [35], similar with WSN430 node, it has sensors like ambient sensor light, atmospheric pressure and temperature, tri-axis accelerometer, magnetometer and tri-axis gyrometer, it supports FreeRTOS operation system, Contiki operation system and Riot operation system.

Figure 3.4(c) is an A8 node, the A8 node is the most powerful node in FIT/IoT-LAB, representative of more advanced devices and different with other nodes that support more higher level operating system like Linux, it also tri-axis accelerometer/magnetometer sensor and tri-axis gyrometer sensor.



(a) Turtlebot2.

(b) Wifibot.

Figure 3.5: The mobile nodes in FIT/IoT-LAB.

Some mobile nodes with predefined trajectories are free to use in FIT/IoT-LAB, each mobile node is embedded on a robot (Figure 3.5(a) and Figure 3.5(b)), Turtlebot2 is an open-source robot, it has an infrared beam which can help robot find their charger position. The Wifibot has infrared sensors and camera, it can mark by a QR code and move forwards.

FIT/IoT-LAB supported various operating systems, especially the five popular IoT operating systems, table 3.2 explains the correspondence between hardware and operating system.

	WSN430	**M3**	**A8**
RIOT	√	√	×
OpenWSN	√	√	×
FreeRTOS	√	√	×
Contiki	√	√	×
TinyOS	√	×	×
Linux	×	×	√

Table 3.2: Operating systems availability [8].

- RIOT (The friendly Operating System for the IoT) is a free, open-source operating system for the IoT, Both embedded devices and common PCs can run the RIOT operating system. Similar with Linux it supports C and C++, but the minimum RAM and ROM is smaller than Linux, the minimum RAM and ROM for Linux is around 1 MB, but for RIOT, the minimum RAM is around 1.5 KB, and the minimum ROM is around 5 KB. At meanwhile, it supports various network stack, such as CoAP at the application layer, UDP at the transport layer, RPL, also has IPv6, ICMP, and 6LoWPAN at the network layer.

- OpenWSN, an open-source implementation of protocols stacks based on IoT standards like the application layer, CoAP, transport layer, UDP and TCP, also include IPv6 and 6LoWPAN.

- FreeRTOS is a real-time kernel; the design of FreeRTOS is small and simple. The core of kernel has only 3 C files. In order to make the code easy to read, port and maintain, most of the code is written in C language, only some functions are written in assembly language. Similar to RIOT, the minimum RAM and ROM for an RTOS kernel is around 6KB to 12 KB.

- Contiki (The Open Source OS for the Internet of Things), Contiki is more focused on small sensor nodes. It pays more attention to communication with nodes in the PAN. Of course, it also has traditional IPv4, IPv6, and TCP, UDP support, CoAP can be used to communicate with the cloud. it also supports the low power protocol 6LoWPAN and RPL. Contiki has a very special simulator, Cooja Network Simulator, which can run many examples, monitor the package and node status of the entire network that allows users to develop without sufficient hardware.

- TinyOS is an open-source operating system designed for low-power wireless devices. The operating system is based on a component-based architecture which allows programs to be updated quickly. The size of minimum RAM and ROM is similar to RIOT, light, and small. The minimum RAM is smaller than 1KB, and the minimum ROM is smaller than 4 KB.

3.4 IoTrain-Lab

IoTrain-Lab, a hands-on IoT training platform using FIT/IoT-LAB testbed. Our platform is open-source, including both fundamental training and security training, as a lightweight, user-friendly training tool.

IoTrain-Lab was developed on the Ubuntu OS, and trainees can use their browser to check the training course webpages via a Moodle learning platform container and use an online Linux environment to do hands-on training on the FIT/IoT-LAB testbed. Training environments are connected with the Moodle container through a client-less remote desktop gateway called Apache Guacamole (see Figure 3.6 for details).

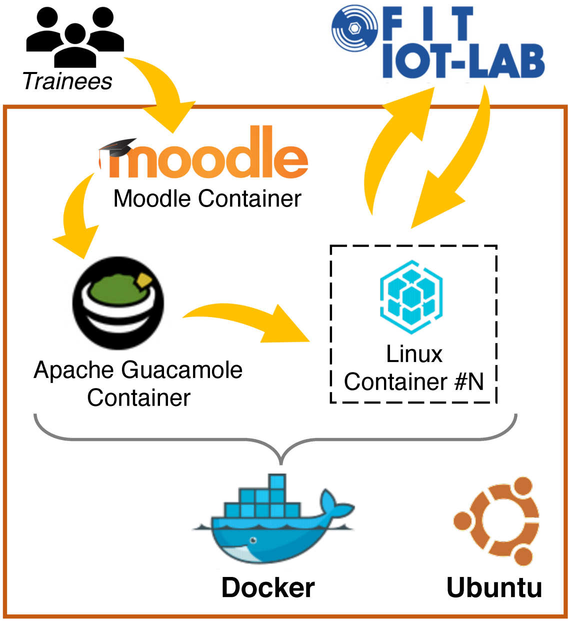


Figure 3.6: Platform architecture.

Comparing with the virtual machine, we use docker container which is fast starting up, small resource usage and high resource utilization, (more details about the container and virtual machine presents at chapter 3.5.2); basically, there are three kinds of containers we used:

- Moodle container, an open-source learning platform designed to provide a learning environment for web-based learning like students and employees.
- Linux container, we chose CentOS as an element, installed Xfce [41] as a lightweight desktop environment, the Xfce is not only fast and low resource consumption, but also visually appealing and user-friendly.

- Apache Guacamole container, it is a clientless remote desktop gateway. It supports standard protocols like VNC, RDP, and SSH [42].

As a user-friendly training platform is not only good enough for trainees but also should handy for instructors. From the point of view of instructors, our platform has two strong points:

- All the tools we motioned in this platform are open-source, IoTrainLab is developed via docker on the Ubuntu OS, the environment can be managed like stop, restart by few command lines.

- Address the time cost issues, we chose docker container instead of the virtual machine, which is using memory efficiently and can start faster (the comparison details are in chapter 3.5.2). Meanwhile, we through docker-compose use YAML files to configure the application' s services, which means the installation procedure is simple as well. Figure 3.7 describes the example to use *docker-compose.yml* and *Dockerfile* to create a CentOS container.

```
## Custom Dockerfile
FROM consol/centos-xfce-vnc
ENV REFRESHED_AT 2019-06-25

# Switch to root user
# install additional software
USER 0

## Install a gedit
RUN yum install -y gedit \
    && yum clean all

## switch back to default user
USER 1000
```

(a) Dockerfile

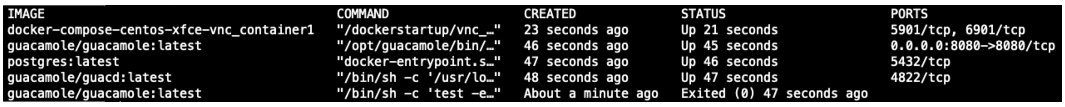
```
version: '3'

services:
  container1:
    build:
      context: .
      dockerfile: Dockerfile
```

(b) Docker-compose.yml

Figure 3.7: Using docker-compose.yml and Dockerfile to create CentOS.

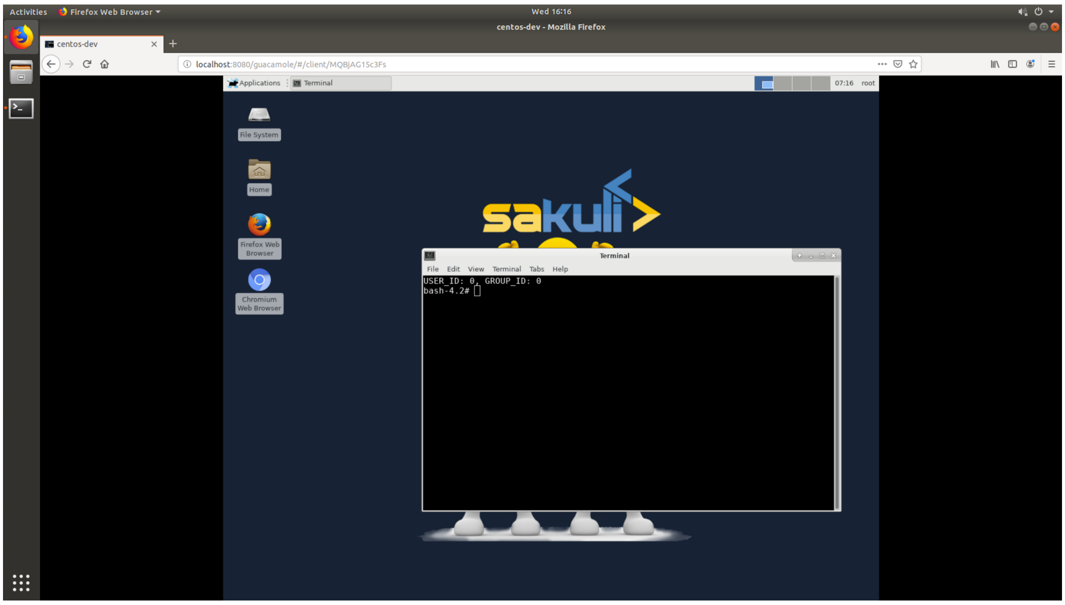
Based on these two files, the instructor is able to use only one command line *docker-compose up -d* to finish the installation. When they complete the installations of Apache Guacamole container, and Linux container which is a CentOS has pre-install VNC and Xfce desktop, the instructors can check using *docker ps -a* to check the currently running containers(Figure 3.8), the items of container ID and name are omitted because of space.



```
IMAGE                                      COMMAND                   CREATED              STATUS                    PORTS
docker-compose-centos-xfce-vnc_container1  "/dockerstartup/vnc_…"    23 seconds ago       Up 21 seconds             5901/tcp, 6901/tcp
guacamole/guacamole:latest                 "/opt/guacamole/bin/…"    46 seconds ago       Up 45 seconds             0.0.0.0:8080->8080/tcp
postgres:latest                            "docker-entrypoint.s…"    47 seconds ago       Up 46 seconds             5432/tcp
guacamole/guacd:latest                     "/bin/sh -c '/usr/lo…"    48 seconds ago       Up 47 seconds             4822/tcp
guacamole/guacamole:latest                 "/bin/sh -c 'test -e…"    About a minute ago   Exited (0) 47 seconds ago
```

Figure 3.8: The running containers.

As the Figure 3.8 showed, if the containers are running as our expected, the instructor can connect Linux container with the clientless remote desktop gateway Apache Guacamole, then, the trainees can use the appointed link to access with Linux container and then get online, with no need for install the configured training environment (Figure 3.9).



```
USER_ID: 0, GROUP_ID: 0
bash-4.2#
```

Figure 3.9: The training environment.

From the point of view of trainees, IoTrain-Lab is an online user-friendly training platform, which can not only help trainees grasp quite complex knowledge, but also inspire their creativity through tutorial learning and hands-on practice. There are just three steps to implement training experiments:

- Use a browser to access the Moodle webpage to get training tutorials;

- Access the online Linux environment;
- Based on the process in tutorials, use the online Linux environment to submit an experiment to the FIT/IoT-LAB testbed, observe and analyze the results.

To create tutorials, instructors need to collect information from various materials such as books, academic papers, videos, international standardization, IoT projects, etc.. (See the details in Figure 3.10).

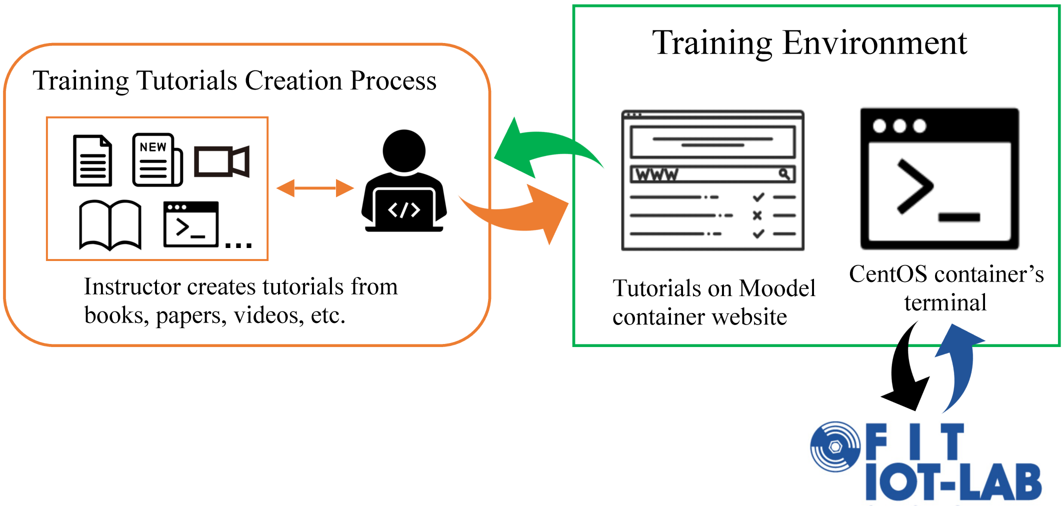

Training Tutorials Creation Process
NEW
Instructor creates tutorials from books, papers, videos, etc.
Training Environment
WWW
Tutorials on Moodel container website
CentOS container's terminal
FIT IOT-LAB

Figure 3.10: Contents creation process.

When the instructors established a tutorial that meets the requirements for trainees, they can release it to the course website.

3.5 Extend tools

This chapter has two sections; the first section introduces the main tool that we used, Docker. Since its release in 2013, it has been highly regarded and considered likely to change the software industry. The second section presents Apache Guacamole which is a gateway service based on Linux that can convert common remote protocols such as RDP (Remote Desktop Protocol), SSH (Secure Shell), VNC (Virtual Network Computing), and Telnet into HTTP (The Hypertext Transfer Protocol). The client can access the remote control via the browser.

3.5.1 Linux container

As we have known, one of the biggest troubles in developing is the environment configuration. Because the personal habits that the user's computer environment is different. When the user wants to use the software, they should guarantee two things at least: the settings of the operating system and the installation of various libraries and components. If some of the old modules or libraries are not compatible with the current environment, then it is a head-scratching puzzlement. As we often say, "It works on my machine." which means other machines are probably not going to run.

In our daily life, if we lost our computer or system breakdown or buy a new machine, we have to do the configurations from the beginning which will take time. So, how can we solve this problem fundamentally? Can we hold the original environment exactly when we install the new software? The answer is yes. We will address two kinds of way for it, using virtual machine and container. Figure 3.11 compares the different structure between virtual machines and containers.

- Virtual Machine

  Virtual machine is a kind of way to solve the problem that we mentioned hereinbefore. It can run another operating system at the current environment, such as running Linux at MacOS, the Linux like a normal file for macOS, it can be deleted without effect when the user does not need it anymore. However, the virtual machine has a few inherent weak points:

  (i) Occupied more resources. Because the virtual machine monopolizes a part of the memory and hard disk space, even if the size of application inside the virtual machine is only few MB, the physical machine still need to preserve hundred MB or GB of memory to support the virtual machine.

(ii) Start slowly. A virtual machine is a complete operating system; it may take a few minutes to start the system.

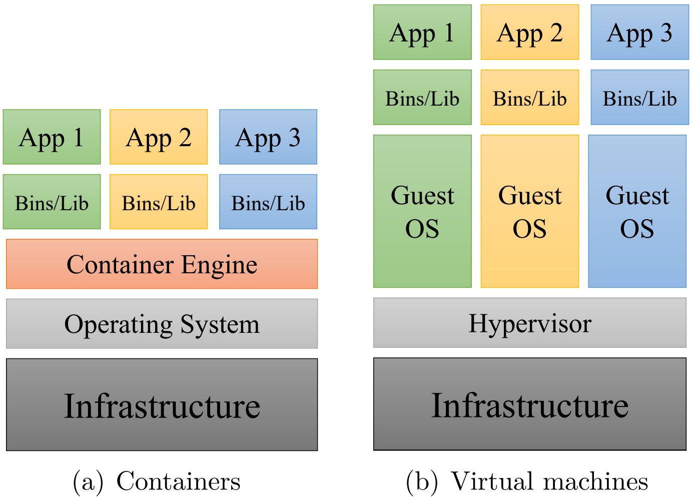
- Linux Containers

  Due to these shortcomings of virtual machines, Linux has developed another virtualization technology: Linux Containers (LXC). Instead of emulating a complete operating system, the function of Linux container is isolating processes. Since containers are process-level, there are many advantages than virtual machines.

  (i) Start fast. The application inside the container is a process of the underlying system because starting a process is faster than starting an operating system; therefore, starting the container is faster than starting a virtual machine.

  (ii) Less resource occupation. The container only occupies the required resources, but for virtual machines will use all resources because it is a complete operating system.

  (iii) Limited size. The virtual machine is packaged via the whole operating system, but for containers only needs to contain the necessary components. Therefore, the size of the container is much smaller than the virtual machine file.

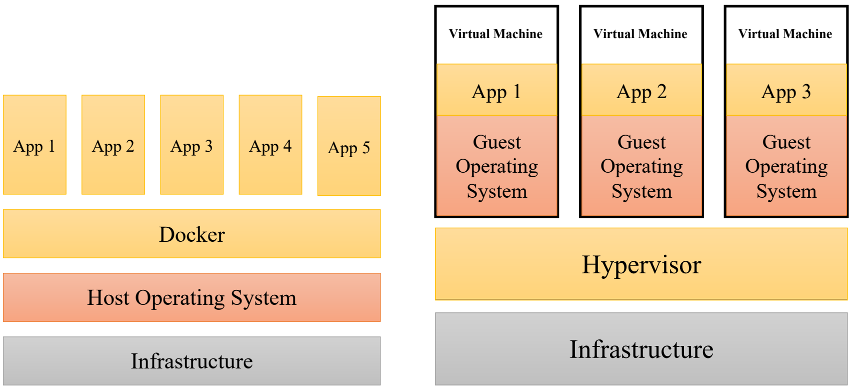


(a) Containers (b) Virtual machines

Figure 3.11: The difference between containers and virtual machines [1].

3.5.2 Docker

Docker is developed in the Go language that as a tool to manage the containers, user can use Docker to create, delete, run the application containers. Currently, it is the most popular Linux container solution. Docker put the application and application related library as a package; a virtual container will be generated if the user runs this package, the program will run inside this virtual container like a real physical machine. Thus, there is no need to mention the environmental issues.



(a) Docker containers

(b) Virtual machines

Figure 3.12: The comparing of docker containers and virtual machines [2].

Docker containers are more portable and efficient than virtual machines. Figure 3.12 presents the different between docker containers which we used for IoTrain-Lab and virtual machines.

3.5.3 Apache Guacamole

We want to be able to access the Linux container remote on the browser, and without install any plugins, the Apache Guacamole is suitable for our needs. Apache Guacamole is a clientless remote desktop gateway [42]. We can access with Linux container from anywhere and anytime, which means as long as trainees are able to access with a web browser (browser needs to support HTML5), they have access with the training environment.

Guacamole is not an independent web application; it made by many components, such as Guacamole server, Guacamole protocol, etc.. More specifically, the web application is the smallest and lightest in the whole project. Most of the functions depend on the underlying components of Guacamole. The architecture of Apache Guacamole is built as Figure 3.13.

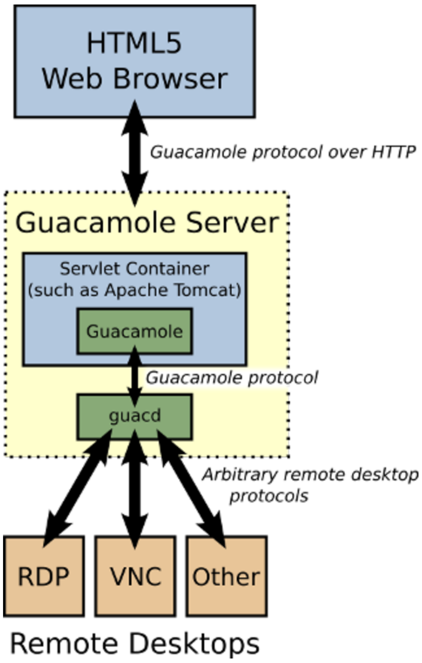


Figure 3.13: The architecture of Apache Guacamole [3].

As we can see from Figure 3.13, there are four parts in the architecture, HTML5 web browser, Guacamole, guacd and remote desktops from top to down. The working flow is blown:

- The user uses their browser to connect with Guacamole Server, the Guacamole Server is written by JavaScript and via Servlet Container such as Apache Tomcat to back the responses and services to the user. Once loaded, the client part will connect to the server by the Guacamole's own defined protocol via HTTP, the Guacamole protocol.

- When the Web application in Guacamole part resolved the Guacamole protocol, the Web application will pass it to the Guacamole's proxy, guacd. More specifically, this guacd proxy is parsing the Guacamole protocol, instead of the user to connect with any number of the remote machine.

- As we mentioned above, the Apache Guacamole can convert common remote protocols such as RDP, SSH, VNC, and Telnet into HTTP, in other words, because of the Guacamole protocol and guacd, the Guacamole client and Web application needless to know which common remote protocols are actually being used.

Chapter 4

Training Content

This chapter has for subsections, address the training content overview first, introduce fundamental training and security training respectively based on the structure. In section 4.2 mainly presents fundamental training from Devices, Network protocols, and Application. Section 4.3 focus on security training through Endpoint Ecosystem, Communication Network Ecosystem, and IoT Service Ecosystem. At last, we explain what kind of contents did we implement.

4.1 Content Overview

As we can see from the Figure 4.1, the GSMA IoT security training is different with other security models, it throughout both service ecosystem part and endpoint part, this model introduces the major components that are required when using production-ready technology. The Communications Network component is inherent in the IoT area, in order to play the purpose of GSMA IoT model, it connected with other ecosystems, IoT Service Ecosystem, and Endpoint Ecosystem.

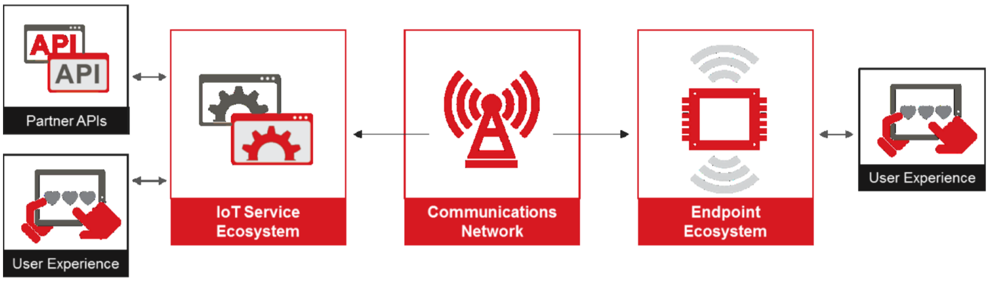
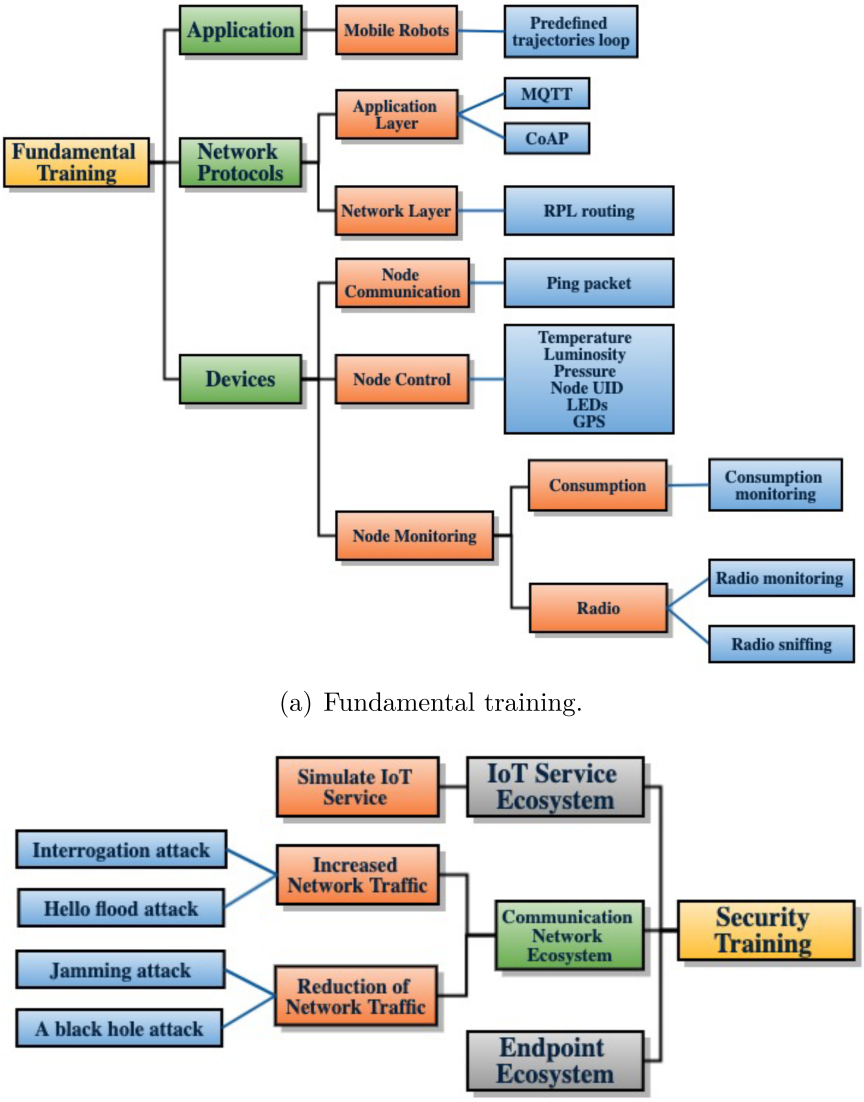


Figure 4.1: GSMA IoT model [4].

As we presented in Chapter 1, IoTrain-Lab has both fundamental training and security training which is a unique feature for IoTrain-Lab. Figure 4.2(b) and Figure 4.2(a) described the classification, structure and details for contents. For Figure 4.2(b), the security training consists of IoT Service Ecosystem, Communication Network Ecosystem, and Endpoint Ecosystem. This classification is based on GSMA IoT security model [4] (Figure 4.1).



(a) Fundamental training.

(b) Security training.

Figure 4.2: Training content overview.

4.2 Fundamental Training

This section mainly introduces the fundamental training, the trainees who did not have considerable knowledge for IoT area or had only a little knowledge about IoT are the targets. This section has three subsections, based on the IoT three-layer architecture we presented at chapter 2.3.2 and the hardware which provided by FIT/IoT-LAB Testbed, we designed this classification, Application, Network Protocols and Devices for fundamental training.

4.2.1 Application

The Application layer is the top of IoT three-layer architecture, it is supported by various IoT protocols, analyze the data formed by the perception layer and feedback to the perception layer for performing specific control functions. It includes controlling the synergy between things and things, the adaptation of things and environment, the balance and cooperation between human and things. For the Application in the fundamental training, it also connected with the sensors in Devices part and network, we combined the significant role of it and the mobile robots in FIT/IoT-LAB (Figure 3.5), designed a corresponding experiment as (Figure 4.3).

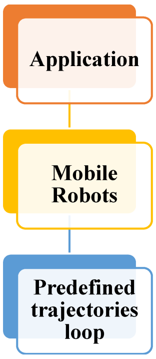


Figure 4.3: Application structures.

This experiment is using mobile M3 node to do a circuit loop, the trainees can learn how to book a node in FIT/IoT Testbed, how to combine and interact with the firmware for it, graph the sensors values, etc..

4.2.2 Network Protocols

The Network Protocols in fundamental training content are organized by Application Layer and Network Layer, for the Application Layer, trainees can do the MQTT and CoAP experiment, for the Network Layer, the RPL Routing experiment has been provided, (see the details at Figure 4.4).

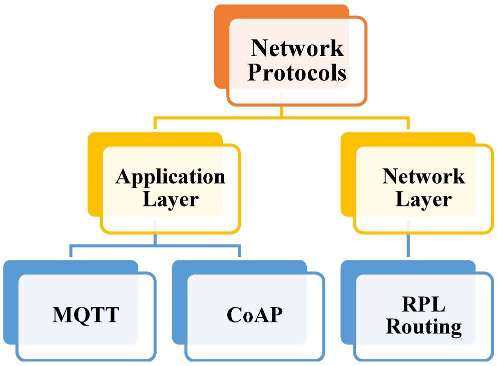


Figure 4.4: Network protocols structures.

The communication between devices, gateways, clouds, and services in the IoT is conducted in accordance with certain communication protocols, such as MQTT [43] (Message Queuing Telemetry Transport), it is a publish/subscribe, simple and lightweight messaging protocol which designed for remote devices and poor network situations. MQTT is an application layer protocol.

CoAP [44] (Constrained Application Protocol) is different with HTTP that runs on the top of TCP (Transmission Control Protocol), CoAP is an application layer protocol that running on the top of UDP (User Datagram Protocol). The CoAP protocol is pretty light, the minimum size of the packet is only 4 bytes. From the human point of view, connect with the Internet looks convenient, however, it is quite difficult for those microdevices to access the Internet. Currently, for PCs, the information exchange is via TCP and HTTP, but for the microdevices, implementing TCP and HTTP protocols is obviously a fallacious demand, thus, In order to allow microdevices to access the Internet, the CoAP protocol was designed.

RPL [45] (Routing Protocol for Low-Power and Lossy Networks) is a routing protocol developed specifically for low power and lossy networks and operated on IEEE 802.15.4, the low power and lossy network is consisting of embedded devices, which have limited power, storage, and processing power, the network connections also have high packet loss rates, low data rates, and instability. The RPL supports three kinds of data flowing ways, point to point, multipoint to point and point to multipoint.

4.2.3 Devices

The perception layer in the IoT three-layer architecture is like skin and facial features of the IoT, used to identify objects, perceive objects, collect information, and automatic control. Similar to the functions of perception layer, the Devices part in fundamental training content is like bricks in the building, it is one of the most basic and significant parts. Considering the different demands for training content and mobilizing trainees enthusiasm, we divided the Devices part into three sub-parts. Node Communication, Node Control, and Node Monitoring, (Figure 4.5).

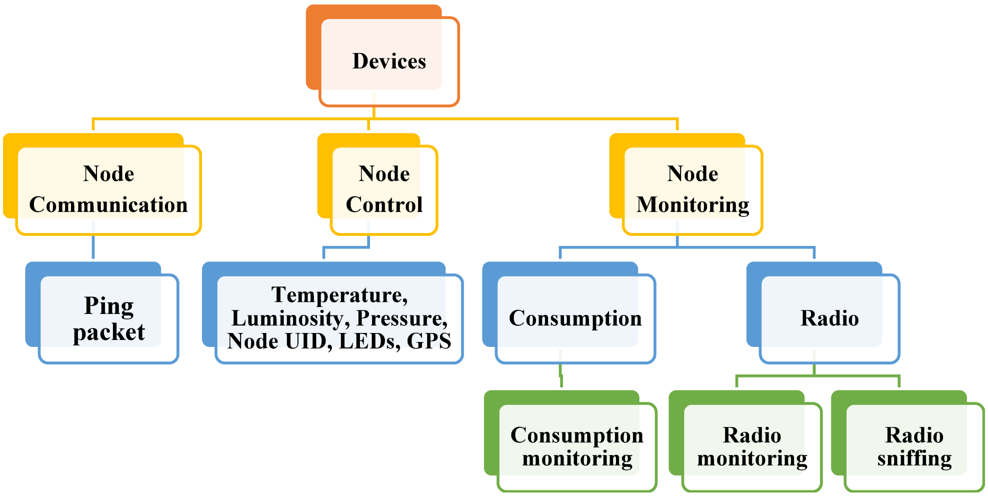


Figure 4.5: Devices structures.

- For the Node Communication part, we introduced the basic communicate experiment, send a ping packet from node A to node B. This experiment can help trainees learn the state of the two nodes communicating and verify the results of the experiment.

- For the perception layer in the IoT three-layer architecture, various sensors used to replace or extend the human senses to complete the perception of the physical world. The experiment which belongs to the Node Control part has the similar functions, we presented the experiment which can help trainees learn how to get environment relevant data such as temperature, luminosity, pressure, GPS, etc..

- Monitor consumption is becoming essential knowledge for someone who would like to have the opportunity to excel in the IoT area. Moreover, with the rapid development of smart things, like smart city, smart grid, smart factory or smart car, monitor consumption from real-time data is an effective and clear way to learn the IoT knowledge. For the Node Monitoring part, we divided it into consumption and radio, each of them has a typical monitoring experiment, for the consumption monitoring, it monitors node consumption via power radio signal and selects power supply from launching an experiment with a firmware that turns on/off red, blue and green LEDs(1Hz period). For the radio categories, a radio sniffing experiment is provided which focus on capturing and analyzing radio communication, it will help trainees capture frames and visualize them in Wireshark.

4.3 Security Training

This section has three subsections, on the one hand, it presents the security training contents via structure, Endpoint Ecosystem, Communication Network Ecosystem, and IoT Service Ecosystem. On the other hand, it explains why did we put gray colors in 4.2(b) which means why did not implement those experiments currently.

4.3.1 Endpoint Ecosystem

As we presented at chapter 4.1, based on the GSMA IoT model, we designed security training contents, at the same time, in the report [46] that wrote by GSMA, it pointed that, there are several inherent weaknesses for the IoT endpoint, such as:

- Low Power Consumption
- Low Cost
- Long-Lived (>10 years)
- Physically Accessible

In the beginning, we divided the Endpoint Ecosystem into two parts, application attack and overwhelm attack based on paper [47]. The application attack means the attacker modifies the firmware or software directly. For the overwhelm attack, the attack might do stimuli for nodes in order to effect its, such as a power consumption increase. It normally requires physical access to the nodes. Due to the FIT/IoT-LAB Testbed is a remote testbed. Besides, the FIT/IoT-LAB Testbed is shared among several users, we are holding the principle that, do not affect others, at the same time, we could not actually access with nodes, thus, we put this Endpoint Ecosystem as gray color in 4.2(b) which means we did not implement these experiments currently.

4.3.2 Communication Network Ecosystem

The Communication Network Ecosystem plays a connecting link between the preceding and the following. At present, there are many IoT protocols, such as Bluetooth, Zigbee, 6LowPAN, Wi-Fi, MQTT, CoAP, etc.. Thus, based on the protocol type, we divided the network protocols in fundamental training into two structures. As we introduced at Figure 4.2(b), for the communication network ecosystem in the security training has been divided into increased network traffic and reduction of network traffic based on the differentiation of impacting for the network in this paper [47] (see the details at Figure 4.6).

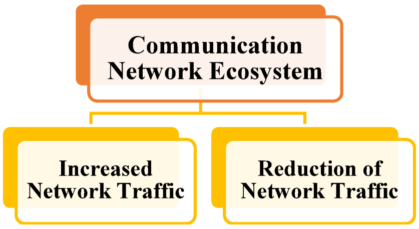

Communication Network Ecosystem
Increased Network Traffic
Reduction of Network Traffic

Figure 4.6: Communication Network Ecosystem structures.

For the increased network traffic, there are some classical examples, such as Interrogation attack and Hello Flood attack, for the Interrogation attack, the attacker ceaseless sends Request to Send packets into the network. For the Hello Flood attack, the attackers injecting “HELLO” packets in order to implement energy consumption.

The Jamming attack and black hole attack can represent the reduction of network traffic, Jamming attack is trying to destroy wireless communication by emitting interference radio frequency. For the black hole attack, it usually happened in the Ad hoc network (Wireless ad hoc network), the malicious node will drop the packet directly instead of forwarding it via the shortest path. This will make the data in the network undeliverable.

For the Flooding attack, it is a attack that introduce packets in the network [47], the propose is to overwhelm the server with massive SYN (Synchronize Sequence Numbers) information and try to exhaust its resources (Figure 4.7). Currently, this Flooding attack experiment which belongs to Increased Network Traffic is available in IoTrain-Lab.

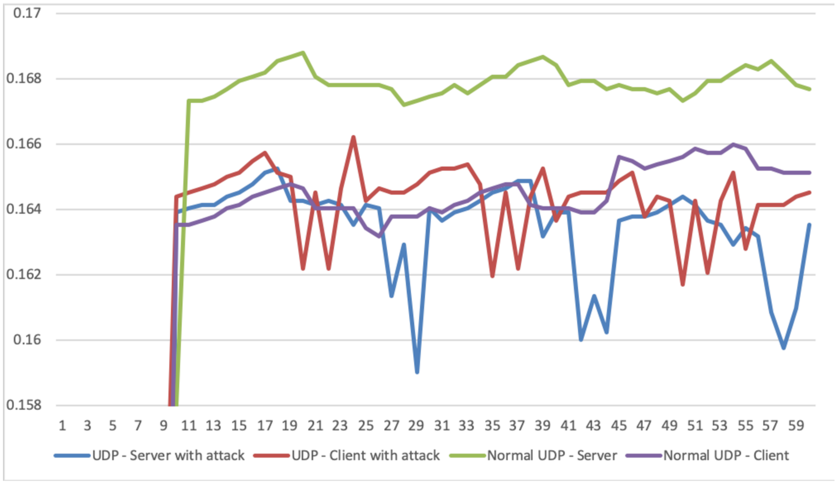
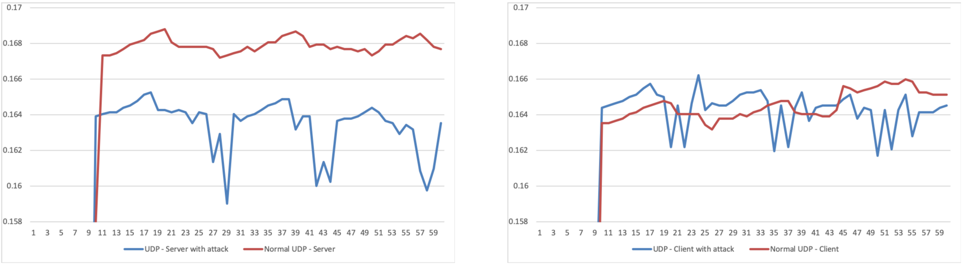


Figure 4.7: Flooding Attack.



(a) The comparison of UDP server with attack and normal server

(b) The comparison of UDP client with attack and normal client.

Figure 4.8: The impact of Flooding Attack for UDP server and client.

As we can see from the Figure 4.7, the power of normal UDP server and client are gentle change over time within a certain range, however, when we flash a attack firmware for both of them, compared to the normal state, the upper and lower peaks of the power vary greatly . Due to we use the public tools FIT/IoT-LAB Testbed for implementing attack, thus, we tested for 60 seconds within the acceptable security range of the nodes.

4.3.3 IoT Service Ecosystem

The GSMA IoT security guidelines for IoT service ecosystems [48] pointed that, currently, IoT productions and services need a service ecosystem in order to provide practical meaning functions and values for users, partner, etc.. Depends on the complexity for the application provided by IoT productions or services, the basic infrastructure might be complex or easy. However, no matter the complexity was, there is one thing which is true that the service ecosystem is a ligament for the whole IoT core technology and communication. Other ecosystems are relied on service ecosystem to access with users, manage and execute other critical tasks, etc..

At the beginning of doing this research, we were planning to set up a simulating for the IoT service via FIT/IoT Testbed nodes, nonetheless, we found well-known platforms like Amazon AWS IoT, Microsoft Azure IoT, all of these platform has powerful functions, supporting equipment management, edge computing, also has various price model. However, currently, these platforms support a limited number of devices and require long-term online equipment to collect enough data for analyzing, graphical data and other operations. As we presented at chapter 3.3, one of the features for FIT/IoT Testbed is open-access, which means sharing nodes with other users, thus, they required the user to reserve nodes before using it.

After a long period of trial and discussion, we decided not to implement this part, for now, we are considering that there are four ways of implementing at the feature. First, find a free and user-friendly IoT platform which can support the nodes from FIT/IoT Testbed. Second, developing or simulating an IoT platform which can support the nodes from FIT/IoT Testbed. Third, finding other open-source, open-access testbed that can be loaded in Amazon AWS IoT, Microsoft Azure IoT or other platforms. Fourth, organizing own IoT testbed that can be supported by various IoT platforms.

Chapter 5

Evaluation

This chapter presents the evaluation of IoTrain-Lab from feature evaluation and user evaluation. For the first section, feature evaluation, we compared IoTrain-Lab with IBM Watson IoT Online Academy [49] which provided by IBM, At the second section, user evaluation, based on 10 questions from the System Usability Scale (SUS) [5], we found five students from Japan Advanced Institute of Science and Technology, gave them a training tutorial based on RPL routing protocol and environment to do the hands-on training, and then let them to answer ten questions for evaluating the IoTrain-Lab.

5.1 Feature Evaluation

This section addresses the feature evaluation via comparing with the Internet of Things Trainer that developed by 3 Rocks Technology which is an engineering training system provider [50].

At chapter 3.3, we presented the major tool that we used, FIT/IoT-LAB Testbed, at chapter 3.4 and chapter 3.5, we introduced the IoTrain-Lab and extend tools for IoTrain-Lab, Docker and Apache Guacamole. There is one benchmark that we selected these infrastructures, open-source. At chapter 4.2, we introduced the training contents for IoTrain-Lab. Essentially, all the training contents were designed for meeting trainees with different educational backgrounds.

For the Internet of Things Trainer that developed by 3 Rocks Technology (Figure 5.1), it provided an environment that user can do the following experiments, at the meanwhile, user can via the Internet or Android App to check the results.

- IoT farm management

- IoT fire alarm
- IoT Internet control toy
- IoT security application
- IoT pet helper

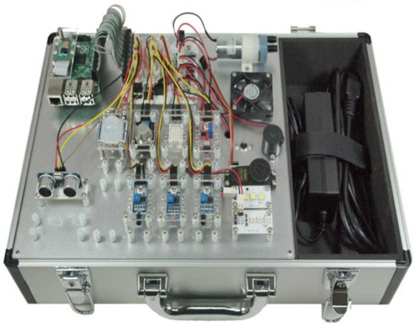


Figure 5.1: System configuration of Internet of Things Trainer.

In order to know the difference, Table 5.1 shows the feature comparison with IoTrain-Lab, IoTrain-System, Internet of Things Trainer and IBM Watson Academy from the various point of view, such as audience, devices form, teaching method, operating system, and price. From the Table 5.1, the advantages of IoTrain-Lab are free to download, abounding nodes supported, well-known IoT operating systems equipped. Form the type of the device point of view, IoTrain-Lab used remote FIT/IoT-LAB Testbed as an infrastructure, the Internet of Things Trainer is more advantageous in the physical control of devices, and the IoTrain and IBM Watson Academy are using virtual devices. Form the training method point of view, due to we implemented the Moodle container that may help the instructors to reduce the workload on statistical results and evaluating effectiveness.

However, comparing with others, the IoTrain-Lab still has some shortcomings, such as the lack of security training tutorials and basic system usage introduction, the infrastructure we used FIT/IoT-LAB Testbed required user to register, in the feature work, we would like to overcome these shortcomings from adding more training tutorials, implementing basic tutorials for trainees without information science knowledge.

	IoTrain-Lab	**IoTrain-system**	**Internet of Things Trainer**	**IBM Watson Academy**
Audience	Everyone	Everyone	Students	Employee or business partner
Pre acquisition knowledge	Low	Low	Secondary or higher education	higher education Medium
Register	No (Register testbed is required)	No	No	Yes
Obtain method	Free download	Free download	Purchase equipment	Online learning
Devices form	Real devices	Virtual devices	Real devices	Virtual devices
Devices type (currently)	13 kinds of boards, total over 1700 nodes	3 kinds of sensor 1 kinds of actuator	8 kinds of sensor 7 kinds of actuator	All devices supported by the IBM Cloud Platform
Devices Mobility	Yes	No	No	No
Content form (IoT correlative)	Moodle course	PDF	Tutorials	Articles, badge, certification, course, tutorials
Teaching method	Self-paced with tutorials	Self-paced with tutorials	Self-paced with tutorials	Instructor-Led Self-paced with labs Self-paced
Training contents	Fundamental training Security training	Fundamental training Security training	IoT design principle How to implement IoT Applications	IBM related
Operating system	FreeRTOS, RIOT, ContikiNG, Zephyr, OpenWSN, TinyOS, Embedded Linux	Contiki OS	Debian, GNU Linux, Arch Linux ARM, RISC OS	System which can be supported by IBM Cloud Platform
Price	Free	Free	Cost	Cost

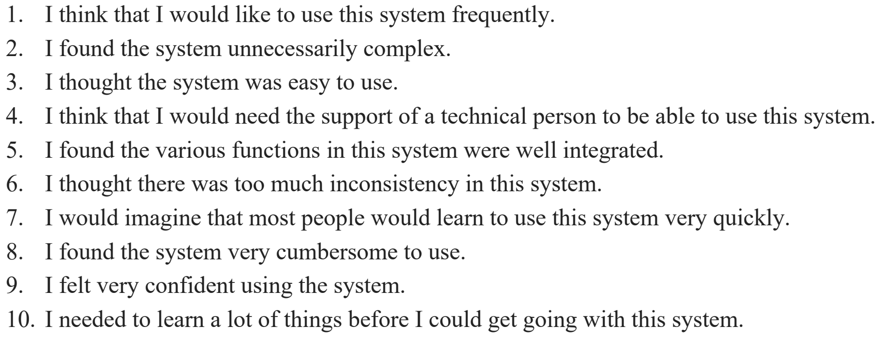
Table 5.1: Feature comparison

5.2 User Evaluation

When we finished the development, we wanted to test the usability. In addition to the qualitative research results, there are quantitative availability questionnaires. These usability questionnaires are standardized, not only can the user experience be scientifically quantified, but developers can also improve and upgrade the system via these questionnaires results.

We found that, there are five well-known scales such as SUS [5] (System Usability Scale), QUIS [51] (Questionnaire For User Interaction Satisfaction), SUMI [52] (Software Usability Measurement Inventory), PSSUQ [53] (Post-Study System Usability Questionnaire) and CSUQ [54] (Computer System questionnaire).

The System Usability Scale (SUS) provides a “quick and dirty” , reliable tool for measuring the usability [5]. It provided ten questions from various aspects like system complex, fluency, user experience, etc., (see the details at Figure 5.2), each question has five range score from Strongly Disagree to Strongly Agree. Odd items are positive descriptions, and even items are negative descriptions. We found five students to do the hands-on training based on RPL routing protocol tutorials and online environment. After completing the experiment, ask them to answer the system usability scale in order to evaluate the overall usability.



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 5.2: System usability scale [5].

There are four main reasons that we use the SUS scale:

- The scale is free on the Internet.
- The whole scale topics are simple and only requires participants to score, thus it is convenient and quick to implement it.
- The measurement result is a score between 0-100, which is easy to understand.

- A number of empirical studies have shown that SUS works better than the others, as paper [6] have shown that the SUS can achieve the fastest results when the sample size is limited, (see details at Figure 5.3).

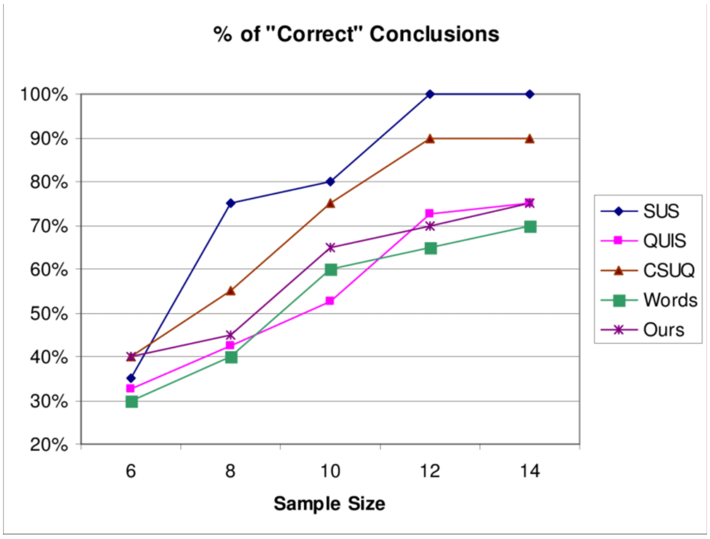


Figure 5.3: A comparison of questionnaires [6].

When the participants finish a series of tasks, they can complete the SUS, as we presented thereinbefore, odd items are positive descriptions, and even items are negative descriptions, thus, the scores of odd items equals "original score - 1" and the scores of even items equals "5 - origin scores" . Once we get the conversed odd items scores and conversed even items scores, it is necessary to add the conversion scores of all items and finally multiply by 2.5 equals a final score of SUS.

In order to more clearly to implement this calculate methodology, we assume that P is equal to the sum of all the positive items scores which means odd items, spi equals the score of each item, when i changes, the value of spi also changes.

Thus, we can get the method for calculating all the positive items scores as follows:

$$P = \sum_{i=1,3,5,7,9} (spi - 1)$$

We assume N is equal to the sum of all the negative items scores which means even items, then, we can get the method as follows:

$$N = \sum_{i=2,4,6,8,10} (5 - spi)$$

Finally, we assume S is equal to the final score:

$$S = (P + N) * 2.5$$

Based on this method, we can get the Table 5.2, it presents the scores of 10 items, final SUS score from five user.

Statement	**Trainee 1**	**Trainee 2**	**Trainee 3**	**Trainee 4**	**Trainee 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
Average	74.5				

Table 5.2: SUS Calculation

As we can see from the Table 5.2, there are five different SUS scores, The SUS score reflects the whole availability, the paper [7] described the acceptability ranges, grade scale and adjective ratings via SUS score, (see the details at Figure 5.4).

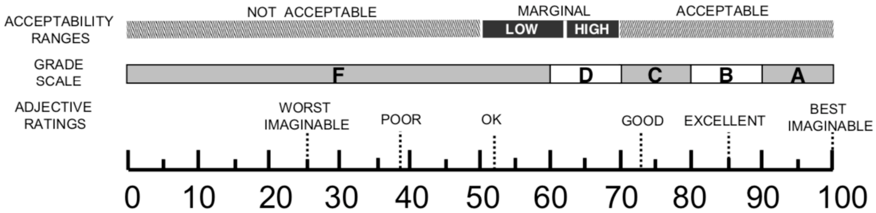


Figure 5.4: SUS Score [7].

Considering the average score is 74.5 and the standard from Figure 5.4, we think that, the IoTrain-Lab is quiet a good and acceptable IoT training platform for both trainees who has cyber-security education background or not, however, due to it involved few knowledge that trainees should know before they official start training, like the composition format of basic Linux command line.

Chapter 6

Conclusion

The idea of this research is coming from we have found that the IoT is becoming more and more common in our lives, with the development of technology, as the IoT continues to infiltrate people's daily lives and the continuous development of artificial intelligence technology, people's demand for the Internet is not limited to the level like "wanting to know new news" , IoT devices can also actively help us improve the quality of life. We want that the objects can "understand our instructions, respond to instructions, and predict what will happen." For example, before we go home, the air conditioner and lighting equipment can start working by predicting the weather temperature and the time. After a long period of recording, the smart speaker can proactively broadcast the news and the latest emails that users care about via learning the user's habit, etc..

However, convenience and humanity are like a double-edged sword. The premise that people feel happy is that IoT is safe enough. Due to the various kinds of cyber-attack is happening day-by-day, it is imminent for users to be aware of security risks and master relevant knowledge. Thus, we did this research, IoTrain, a hands-on IoT security training platform using IoT testbeds, we designed both fundamental and security training contents for trainees with different educational backgrounds, because it has the characteristics of quick installation, convenient management, fewer system resources occupied and low capital requirements, etc., the enterprises and universities can use IoTrain as an easy-to-use IT online training platform.

Having said that, although IoTrain-Lab has great strongpoints, the IoTrain-Lab is not perfect either, we considered the IoTrain-Lab as a training platform has three limitations:

- For this research, we did a comparison of currently existing IoT testbed and chose the first candidate FIT/IoT-LAB Testbed as an infrastructure to do our training for protocols and applications. During the using period, there have been some force majeure factors, such as time delay, device occupied, etc.. At the meanwhile, this research is only using one testbed, which has limitation for sensors type and function. In future work, we would like to find more open-source, user-friendly IoT testbeds in order to provide state-of-the-art, assorted training.

- As we presented at chapter 5.2, we found five students with different education backgrounds, during the evaluation period, we have received a variety of evaluations and opinions. For the students without information science relevant background, a very important point is that they want more introduction training about the basic knowledge, such as the Linux operating system. Thus, we think this is the second limitation for IoTrain-Lab, in the feature work, we would like to add more basic, suitable training experiments for the trainees without information science relevant background.

- Current stage, from the point view of instructors, before they create the training contents, the instructor need to collect accurate information from books, papers, videos, etc., and then they can make new training contents which are manual and time-consuming, thus, we considering that, in the feature work, we would like to add new developing technology like deep learning and natural language processing in order to handle this labor cost. From the point of view of trainees, learning from books at school or from a training platform is a kind of crusted studying way. In the feature work, we would like to combine learning and technology such as using AR/VR technology for three-dimensional learning to have fun while studying.

Bibliography

[1] “Containers vs. virtual machines (vms): What’s the difference?” https://blog.netapp.com/blogs/containers-vs-vms/, accessed July 24, 2019.

[2] “Comparing containers and virtual machines,” https://www.docker.com/resources/what-container, accessed July 24, 2019.

[3] “The implementation and architecture of guacamole,” https://guacamole.apache.org/doc/gug/guacamole-architecture.html#web-application, accessed July 24, 2019.

[4] “Gsma iot security guidelines and iot security assessment,” https://www.gsma.com/iot/wp-content/uploads/2019/04/CLP.11-v2.1.pdf, accessed July 28, 2019.

[5] “System usability scale,” https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html, accessed July 28, 2019.

[6] T. S. Tullis and J. N. Stetson, “A comparison of questionnaires for assessing website usability,” in *Usability professional association conference*, vol. 1. Minneapolis, USA, 2004.

[7] A. Bangor, P. Kortum, and J. Miller, “Determining what individual sus scores mean: Adding an adjective rating scale,” *Journal of usability studies*, vol. 4, no. 3, pp. 114–123, 2009.

[8] C. Adjih, E. Baccelli, E. Fleury, G. Harter, N. Mitton, T. Noel, R. Pissard-Gibollet, F. Saint-Marcel, G. Schreiner, J. Vandaele *et al.*, “Fit iot-lab: A large scale open experimental iot testbed,” in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*. IEEE, 2015, pp. 459–464.

[9] K. Ashton *et al.*, “That ‘internet of things’ thing,” *RFID journal*, vol. 22, no. 7, pp. 97–114, 2009.

[10] D. Evans, “The internet of things how the next evolution of the internet is changing everything,” https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf, accessed June 23, 2019.

[11] “Gartner says worldwide iot security spending will reach $1.5 billion in 2018,” https://www.gartner.com/en/newsroom/press-releases/2018-03-21-gartner-says-worldwide-iot-security-spending-will-reach-1-point-5-billion-in-2018, accessed June 23, 2019.

[12] “The top 10 iot segments in 2018 −based on 1,600 real iot projects,” https://iot-analytics.com/top-10-iot-segments-2018-real-iot-projects/, accessed July 8, 2019.

[13] “Iot market in japan: Iot use cases, players, market outlook to 2022,” https://www.udr-inc.com/ /iot-market-in-japan/, accessed July 8, 2019.

[14] “国内 iot 市場テクノロジー別予測を発表,” https://www.idcjapan.co.jp/Press/Current/20180912Apr.html, accessed July 8, 2019.

[15] “Internet security threat report,” https://www.symantec.com/content/dam/symantec/docs/reports/istr-23-2018-en.pdf/, accessed June 23, 2019.

[16] “Cisco 2018 annual cybersecurity report impacts on public sector,” https://www.noacsc.org/wp-content/uploads/2018/05/Cisco2018AnnualCybersecurityReportImpactsOnPublicSector.pdf/, accessed June 23, 2019.

[17] Towdium. (2017) 国内 iot（internet of things）セキュリティ製品市場予測を発表. [Online]. Available: https://www.idc.com/getdoc.jsp?containerId=prJPJ44480918

[18] “Chinese proverbs,” https://en.wikiquote.org/wiki/Chinese_proverbs/, accessed June 23, 2019.

[19] R. Beuran, C. Pham, D. Tang, K.-i. Chinen, Y. Tan, and Y. Shinoda, “Cytrone: An integrated cybersecurity training framework,” 2017.

[20] R. Minerva, A. Biru, and D. Rotondi, “Towards a definition of the internet of things (iot),” *IEEE Internet Initiative*, vol. 1, pp. 1–86, 2015.

[21] R. Mahmoud, T. Yousuf, F. Aloul, and I. Zualkernan, “Internet of things (iot) security: Current status, challenges and prospective measures,” in

2015 10th International Conference for Internet Technology and Secured Transactions (ICITST). IEEE, 2015, pp. 336–341.

[22] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, “A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications,” *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1125–1142, 2017.

[23] M. Leo, F. Battisti, M. Carli, and A. Neri, “A federated architecture approach for internet of things security,” in *2014 Euro Med Telco Conference (EMTC).* IEEE, 2014, pp. 1–5.

[24] “Avast、日本の 3 世帯に 1 世帯が脆弱なスマートホームデバイスを所有していることを明らかに,” https://press.avast.com/ja-jp/avast_smart_home_repor_2019_ja, accessed July 12, 2019.

[25] “Owasp internet of things project,” https://www.owasp.org/index.php/OWASP_Internet_of_Things_Project, accessed July 12, 2019.

[26] K. Tan, D. Wu, A. J. Chan, and P. Mohapatra, “Comparing simulation tools and experimental testbeds for wireless mesh networks,” *Pervasive and Mobile Computing*, vol. 7, no. 4, pp. 434–448, 2011.

[27] G. Z. Papadopoulos, K. Kritsis, A. Gallais, P. Chatzimisios, and T. Noel, “Performance evaluation methods in ad hoc and wireless sensor networks: a literature study,” *IEEE Communications Magazine*, vol. 54, no. 1, pp. 122–128, 2016.

[28] G. Z. Papadopoulos, A. Gallais, G. Schreiner, E. Jou, and T. Noel, “Thorough iot testbed characterization: From proof-of-concept to repeatable experimentations,” *Computer Networks*, vol. 119, pp. 86–101, 2017.

[29] “Fed4fire+,” https://www.fed4fire.eu, accessed July 1, 2019.

[30] “Geni,” https://www.geni.net/about-geni/what-is-geni/, accessed July 1, 2019.

[31] “Wisebed,” http://www.smartsantander.eu/index.php/related-projects/item/126-wisebed-wireless-sensor-network-testbeds, accessed July 1, 2019.

[32] “Smartsantander,” http://www.smartsantander.eu, accessed July 1, 2019.

[33] “The fit consortium,” https://www.iot-lab.info/about-us/, accessed July 1, 2019.

[34] “The map of geni,” https://www.geni.net/about-geni/geni-maps/, accessed July 31, 2019.

[35] “Various hardware platforms available,” https://www.iot-lab.info/hardware/, accessed July 1, 2019.

[36] “Freertos,” https://www.freertos.org, accessed July 21, 2019.

[37] “Contiki,” http://www.contiki-os.org, accessed July 21, 2019.

[38] “Riot,” https://riot-os.org, accessed July 21, 2019.

[39] “Tinyos,” http://www.tinyos.net, accessed July 21, 2019.

[40] “Openwsn,” https://openwsn.atlassian.net/wiki/spaces/OW/overview, accessed July 21, 2019.

[41] “Xfce desktop environment,” https://www.xfce.org/, accessed July 17, 2019.

[42] “Apache guacamole,” https://guacamole.apache.org/, accessed July 17, 2019.

[43] “Mqtt,” http://mqtt.org, accessed July 29, 2019.

[44] “Coap,” https://coap.technology, accessed July 29, 2019.

[45] T. Winter, “Rpl: Ipv6 routing protocol for low-power and lossy networks,” 2012.

[46] “Iot security guidelines endpoint ecosystem,” https://www.gsma.com/iot/wp-content/uploads/2019/04/CLP.13-v2.1.pdf, accessed July 30, 2019.

[47] A. Diaz and P. Sanchez, “Simulation of attacks for security in wireless sensor network,” *Sensors*, vol. 16, no. 11, p. 1932, 2016.

[48] “Iot security guidelines for iot service ecosystems,” https://www.gsma.com/iot/wp-content/uploads/2019/04/CLP.12-v2.1.pdf, accessed July 30, 2019.

[49] “Ibm watson iot online academy,” https://www.iot-academy.info/index.php/en-us/, accessed July 28, 2019.

[50] “3 rocks technology,” https://www.3rockstech.com/index.php/training-systems/internet-of-things/357-internet-of-things-trainer, accessed July 30, 2019.

[51] “Questionnaire for user interaction satisfaction,” https://www.cs.umd.edu/hcil/quis/, accessed July 29, 2019.

[52] “Software usability measurement inventor,” http://sumi.uxp.ie/about/whatis.html, accessed July 29, 2019.

[53] J. R. Lewis, “Psychometric evaluation of the post-study system usability questionnaire: The pssuq,” in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 36, no. 16. SAGE Publications Sage CA: Los Angeles, CA, 1992, pp. 1259–1260.

[54] ——, “Ibm computer usability satisfaction questionnaires: psychometric evaluation and instructions for use,” *International Journal of Human-Computer Interaction*, vol. 7, no. 1, pp. 57–78, 1995.