Simulation Modelling Practice and Theory 73 (2017) 43–54

Contents lists available at ScienceDirect

Simulation Modelling Practice and Theory

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

Using virtual environments for the assessment of cybersecurity issues in IoT scenarios

Angelo Furfaro[a,*], Luciano Argento[a], Andrea Parise[b], Antonio Piccolo[a]

[a] DIMES, University of Calabria – P. Bucci 41C, 87036 – Rende (CS), Italy
[b] Open Knowledge Technologies srl – Piazza Vermicelli, 87036 – Rende (CS), Italy

ARTICLE INFO

Article history:
Available online 29 October 2016

Keywords:
Virtual environments
Internet of Things
Cybersecurity
Agent-based simulation

ABSTRACT

Internet of Things (IoT) has been forecast as the next main evolution in the field of Information and Communication Technology. Recent studies confirm a steep and constant increase of the diffusion of smart objects, which are capable to interact with the real world and to communicate and gather information autonomously through Internet connections. In the rush of devising and releasing to the market the next killer application for the IoT domain, critical issues regarding the cybersecurity risks are currently overlooked. Because security, like most software qualities, is not an orthogonal requirement, i.e. one that can be satisfied by adding some features at any development stage, this is going to have a very costly impact, potentially leading to failure, on the technologies currently developed for the IoT domain. In order to cope adequately with such issues, IoT applications need to be designed to be secure since their inception and suitable security assessment tools should be made available. This paper describes an approach based on the exploitation of virtual environments and agent-based simulation for the evaluation of cybersecurity solutions for the next generation of IoT applications in realistic scenarios. The effectiveness of the approach is shown by considering a concrete case study involving the cooperation of real and virtual smart devices inside a virtualized scenario where security issues are first evaluated and then handled.

1. Introduction

Since its beginning, Internet has proven to be a very changing and evolving environment. Born as a simple network of computers, over time it evolved its shape and its features to become the complex infrastructure of today which allows the huge flow of daily worldwide information sharing and that enables the enormous amount of digital services which are pervasive to almost all types of business. Internet applications are involved in a large and ever growing number of aspects of common people life. Most of these applications were not foreseen since its inception and a lot of issues had to be addressed as they spontaneously appeared.

During the last few years, Internet applications are still increasing and, in particular, an outstanding number of highly heterogeneous networked objects (things), many of which characterized by small size and low power consumption [1] are

* Corresponding author.
E-mail addresses: a.furfaro@dimes.unical.it, a.furfaro@unical.it (A. Furfaro), l.argento@dimes.unical.it (L. Argento), andrea.parise@okt-srl.com (A. Parise), a.piccolo@dimes.unical.it (A. Piccolo).

http://dx.doi.org/10.1016/j.simpat.2016.09.007
1569-190X/

becoming part of Internet, e.g. implantable medical devices, smart thermostats, smart meters, or any object that has the ability to transfer data over a network.

This trend has been widely recognized as the next main step in the evolution of Internet which is commonly referred to as the Internet of Things (IoT). With respect to traditional Internet sources of information, in the IoT scenarios data come from the physical world through the sensors installed on smart devices, thus widening the range of possible applications, e.g. involving the processing of environmental data and make intelligent decisions on the surrounding environment. IoT is becoming a bridge between the physical and the digital world by including smart objects which interact with the physical environment without direct human intervention [2].

IoT is showing the potential for impacting several domains, ranging from personal to enterprise environments [3]. Examples of domains and possible applications include, but are not limited to, smart cities, for lowering energy costs and reducing pollution, and smart homes, for which energy companies are building systems to increase energy savings and safety.

Despite the goals of IoT applications are directed to improve most aspects of both business and common people's life, such emerging technology is also becoming an increasingly attractive target for cybercriminals. The more are the Internet connected devices the more are the potential attack vectors and the vulnerabilities that malicious entities may exploit.

Estimates on the number of devices that will be connected to the Internet by 2020 range from 20.8 billion [4], to 30 billion [5] devices. According to Gartner, "by 2020, addressing compromises in IoT security will have increased security costs to 20 percent of annual security budgets, from less than one percent in 2015" [6]. Unfortunately, as reported by [7], cybersecurity risks have received little attention up to now. It seems that IoT is retracing the same path that the Internet undertook during its evolution: most of the attentions are focusing on the technologies needed to achieve the desired functionalities, neglecting the aftermath of the security issues that are going to arise.

An important factor to consider is the presence of manufacturers that lack prior experience with networked devices: in an attempt to place into the market their devices and get the newest and attractive functions at the lowest cost, as quickly as possible, they end up neglecting the design and implementation of security features for hardware and software.

It is of utmost importance giving to security a high priority during the development process of IoT, otherwise, in the near future, the number of security risks for consumers and businesses will increase exponentially, leading to disastrous situations for both sides. Therefore, security should not be an artifact added at the end of the development, but it must be an integral part of the entire process. Consequently, the devices placed in the markets, should be equipped with built-in security mechanisms and ensure greater protection for their users.

To address the security vulnerabilities of IoT devices created so far, researchers are focusing on the evaluation of security properties [8]. The goal of this analysis is to identify and understand the security issues of currently deployed devices and help manufacturers to solve the detected problems, by providing them with guidelines and recommendations for improving the security of future software updates and/or version of the devices. Towards this objective, computer simulation techniques along with novel cloud based virtualization platforms represent a very good combination for achieving suitable cybersecurity analysis and assessment platforms. Virtual environments are systems in which realistic scenarios can be reproduced, by exploiting computer and network virtualization technologies and agent-based simulation [9,10]. They find applications in many domains including military, medical, educational and recently also in cybersecurity [11].

This paper illustrates how virtual environments can be a valuable tool to assess security properties and discover vulnerabilities of IoT devices, in realistic scenarios. Specifically, the SmallWorld platform is proposed for the development of intelligent virtual environments in which the agent paradigm is used to simulate malicious and legal behaviors, both of machines and human beings. SmallWorld has being developed to be scalable by design. It introduces an abstraction layer and a set of API which make it able to run on different hypervisor technologies ranging from single machine solution (e.g. VirtualBox) to state of art cloud solution.

The SmallWorld effectiveness in the assessment of IoT cybersecurity concerns is shown through a case study in the context of smart home applications.

The rest of the paper is organized as follows. Section 2 gives an overview of the related work. Section 3 discusses the main security issues affecting IoT technologies and devices as they are currently developed, implemented and deployed. Section 4 describes the use of virtual environments as a security analysis assessment tool. Section 5 presents a case study involving smart home applications. Finally, Section 6 concludes the paper.

2. Related work

This section overviews the various approaches for the assessment of security concerns in IoT systems as described in the literature.

A model-based security toolkit, SecKit [8], has been proposed to enable the protection of user data by supporting specification and efficient evaluation of security policies. SecKit is integrated into a generic management framework for IoT devices. It has been designed to support the modeling of IoT systems and to specify, in an integrated way, security requirements, usage control policies, threat scenarios and trust relationships. These issues are addressed by means of meta models and a policy rule language. In particular, the adoption of trust models enables the specification of trust relationships, of various types, by which governing the trust relationships in the IoT interactions. The use of SecKit was experimented in [8] under two scenarios, regarding smart home and city.

A framework for modeling and assessing security in the IoT is described in [12]. Specifically, the objectives of the framework are to graphically represent all possible attack paths in an IoT network, whose configuration is provided as input by a security decision maker, in order to evaluate the effectiveness of possible defense strategies. The use of the framework has been experimented in the domain of pervasive healthcare monitoring and of environment monitoring. However, its main limitation is due to the fact that it requires a sensor networks of identical nodes which is very unlikely to occur in real IoT settings. In addition it is not able to take into account the mobility of devices.

Another general security assessment framework for IoT services is discussed in [13]. The proposed approach uses integrated fuzzy multi-criteria decision-making methods. It exploits a combination of a fuzzy analytic network process (ANP) and of the fuzzy decision-making trial and evaluation laboratory (DEMATEL). The former is used to assign a weight to each IoT security requirement, the latter is employed to derive cause-and-effect interrelationships between the security criteria. The framework aims at handling both qualitative and quantitative security criteria.

Security issues regarding the use of IoT in the field of eHealth applications are considered in the work described in [14] where a framework for the assessment of context-aware adaptive security solutions is applied in this context. A set of IoT-eHealth scenarios is provided and it is considered for the evaluation of the approach. Further, the framework employs linear and logarithmic approaches to assess and quantify the security and QoS requirements of the applications, in an adaptive security system. The evaluation methodology is based on a comparison between results from laboratory experiments and simulations and the assessment by human observers. The work presented in [14] is complemented by that discussed in [15], where, in addition to QoS and security requirements, user preferences and device capabilities are also taken into account.

A metric-based approach to assess the security level of IoT connected Critical Infrastructures (CI) is proposed in [16]. The authors introduce a set of suitable security metrics on the basis of which the satisfaction of security requirements is checked. The metrics are employed to define Service Level Agreements (SLAs) in which the requirements and the penalties that must be applied in case of violations are defined. The approach is evaluated in the context of a financial infrastructure, however, the approach is generic and can also be applied to other CIs. A discussion on the specific security issues related to IoT applications for the Smart Grid is reported in [17].

While the above described approaches have their merits in the view of design of secure IoT applications, they are mostly based on theoretical models from which evaluation framework are derived and, as a consequence, they tend to overlook some practical details which may hide serious security holes that can be lately discovered only when IoT systems are put into operation. In order to fill this gap, the availability of a platform allowing to reproduce in a realistic way (part of) an IoT infrastructure accounting for low level operation details (e.g. operating system and employed library versions, specific hardware and software components, network topology, firewall rules, etc.) is of critical importance. To the best of our knowledge, this paper is the first proposing the combined use of virtual environments, agent-based simulation and real devices in order to allow accurate evaluation and assessment of realistic IoT deployment scenarios which may involve complex networking infrastructures.

3. Security concerns

IoT has brought interesting opportunities both for consumers and businesses, however it came together with vast repertoire of new security challenges. IoT technologies are *embedded* into and extend the Internet ecosystem and, as a consequence, they inherit all the Internet related security problems and pose new specific issues. Because of the pervasive nature of IoT devices and applications, these security problems are of a greater importance and, in some cases, tend even to become critical. To cite one example, a group of researchers, in 2008, showed how it was possible to extract personal information from a pacemaker or even to threat the life of a patient by altering the behavior of the device [18].

Similarly to the Internet, the IoT can be subject to a high number of threats, such as attacks that target diverse communication channels, physical threats, denial of service, identity fabrication, and others [19]. Unlike the Internet, in the IoT the attack surface increases exponentially given the high number of interconnected devices. The current state of IoT is also characterized by the absence of standards and the extremely heterogeneous nature of the devices, in the hardware, software and adopted communication protocols (Wi-Fi, Z-Wave and ZigBee, to name a few). All these conditions introduce considerable complexity into the design process of the general security solutions. Moreover, in a typical Internet scenario the connected resources, e.g. desktop computers, have enough computing power to run software tools, e.g. antivirus, that can protect them from some threat sources. In the IoT instead, the devices have limited resources and because of this the use of existing technologies such as antivirus is often impracticable.

Having said that, it is clear how important it is to pay special attention to the security topic in the IoT. Unfortunately, it seems that it is not the case. The producers are devoting much of their attention to the development of capabilities and technologies with the objective of achieving long-awaited services and get a rapid spread in the IoT market.

In a study conducted by HP [7], it was found that the security topic in the IoT has had a very weak presence in industrial and academic conferences in 2014 and 2015, with respect to other application domains.

It would seem that the IoT is retracing the same steps of the Internet in its infancy. Decades ago, when the Internet was going through its early stages of evolution, those who devoted themselves to its design and development were focused on technical issues with the aim of being able to transfer information quickly and reliably. These people were shortsighted about information security, they primarily took into account military threats, but they failed to understand that the same

Internet users a day might become threats. This led to a situation in which it was necessary to introduce ex post information security solutions in response to a very high number, with variable severity, of threats.

It is desirable to undertake a change of direction, i.e. by stopping to neglect the security aspect and by investing resources to design the devices taking into account the possible security issues that may afflict them in the future. In this way, devices with built-in security will be introduced in the market, offering a better protection to the customers.

The following subsections provide more information on the security landscape of IoT. Specifically, they describe what are the main threats for the IoT, what are the motivations that push malicious users to act, what types of attack patterns can be observed and some examples of exploits.

3.1. Threat sources

IoT devices manage a huge quantity of information, related e.g. to lifestyle habits of a consumer, and they are capillary distributed in every industry [20]. This aspect is the main reason why the IoT security is threatened. Criminals, government entities, and hacktivists are just few examples of actors who harbor interests with respect to these data. For example, a group of criminals might be interested in stealing sensitive information by hacking specific devices. In this scenario it becomes potentially easier also to observe cases where a person, for personal reasons, may disturb the daily life of another person, by altering the normal functioning of the devices installed in the victim's home.

Three main categories of malicious entities threating IoT can be identified [21]: *i)* external attackers, *ii)* malicious users and *iii)* bad manufacturers.

An *external attacker* is an entity that does not have permission to access a system or a device. He usually remotely target a device (or set of devices) by exploiting its vulnerabilities and he can have various goals, e.g. stealing sensitive data, causing malfunctions or financial damages.

A *malicious user* is identifiable as the owner of a device from which he wants to extract data relating to secrets of a manufacturer, or gain access to features not accessible to the user. One of its objectives could be to sell secrets to a third party, e.g. in the case of a former employee of a company, driven by resentment.

As malicious users may be interested in obtaining sensitive information from a manufacturer of a certain device also the opposite situation could happen. A *bad manufacturer* might be interested in gathering information about its general users or about a specific user's habits. To achieve this result it could deliberately introduce security holes, by means of which it is possible to gain access to user data, violating his privacy. A manufacturer could also be interested in seeking information on other IoT devices or it might even try to attack other devices, produced by competitor firms, in order to damage their reputation.

3.2. IoT exploit scenarios

This section reports two cases of IoT device exploits in order to make clear the impact that the presence of vulnerabilities in smart devices can have on people's lives. During the 2015 edition of the Black Hat USA security conference, Miller and Valasek showed how they were able to compromise a smart car, specifically a Chryslers Jeep [22]. The two researchers explained that there were two ways to perpetrate the attack. In one case, the victim must had to be subscribed to the wireless connection service from the manufacturer. They found out how the Wi-Fi password is generated, i.e. based on the default system time plus a few seconds due to the boot procedure of the head unit. The date corresponded to `January 01 2013 00.00 GMT`, and in the specific case study to `00.00.32 GMT`. The number of combinations to be generated was small, therefore, little effort were required to guess the password. Once a connection was established with the Jeeps head unit, it was possible to find a way to hack the multimedia computer, which runs on a Linux operating system. They managed to take control of the head unit of the system by exploiting some pretty guessable flaws in the software. Of course, not all consumers can be interested in signing the service offered by the manufacturer. In the second case study, they shown that it was possible to obtain control of the system, leveraging the connection that all head units had with the Sprint cellular network. More details can be found in the article published by Miller and Valasek.

Any type of device may be part of the IoT, including dolls. In 2013 Mattel has put on the market *Hello Barbie*, a doll which uses Wi-Fi to transmit what children say to it to remote servers that process the speeches and build suitable replies. Researchers showed that the doll had few insecurities. Studies conducted on Android and iOS applications associated with the doll, revealed the presence of serious defects by which an attacker is able to eavesdrop on communications between the cloud server and the doll [23]. Furthermore it was showed that the application will automatically connect to any Wi-Fi network whose name includes "Barbie". These flaws were exploited to gain access to system information, Wi-Fi network names, internal MAC addresses, account IDs and MP3 files. Furthermore, these data can be used to find someone's house and access personal information.

3.3. Attack vectors/models

In the world of IoT, old and new attack patterns arise. When IoT will have reached full maturity, smart devices will be everywhere, e.g. in our homes and offices. This will allow an attacker to be able to get physical access to a device, i.e. the highest level of access. Although it may seem hard to believe, physical access will be a plausible attack vector. Just think

of a guy who still has access to the home of a former girlfriend, he will have access to the devices and try to reconfigure them to spy on the movements of the victim. The attacker could exploit the physical access to capture a device and extract the information contained in it or alter its configuration, an attack pattern called node capture [24]. It would therefore be possible to reset the device in an attempt to restore its original settings, or install a custom Secure Socket Layer certificate for directing traffic to a server under his control. It is also conceivable that an attacker compromises a device in his possession and resells it to spy on other people. This new attack vector also allows to conduct a Denial of Service attack. Indeed, if the attacker is able to access a device, it might make it unusable, by destroying it.

The above mentioned Denial of Service (DoS) attack is another serious threat to IoT. A DoS attack is defined as any event that diminishes or eliminates a network capacity to perform its expected function, degrading the quality of the services offered to its users [25]. These attacks can be initialized from remote places with mere commands, combined with advanced tools. Distributed DoS attacks may also be performed, which are more effective in exhausting the networks' resources. In the IoT one of the primary objectives for this threat is the wireless communication infrastructure. By using attacks like Jamming [26], which is a special case of DoS attacks which interferes with the radio frequencies used by sensor nodes, an attacker may prevent communications between smart devices, making it impossible for exchanging information, a vital aspect in the IoT.

Another longtime threat reoccurring in the IoT is the malware spread. The first malware in the Internet of Things was discovered by Symantec in 2013, and was named `Linux.Darlloz` [27]. Malwares are a very powerful means to compromise a device. It can be exploited to reach another device that contains the data of interest. Unfortunately the limited resources available to the devices, make it hard to deal with this threat. It is not possible to use tools like antivirus, in order to recognize malwares in real time, because they would require an unsustainable strain on the device.

Although the focus on IoT security is not the same as that given to the development of the technologies needed to implement the desired services, it is possible to observe a number of works that propose solutions to various problems. To cite one example, the work published in [28] copes with the presence of loopholes in device security and data integrity, by proposing an access control and authentication mechanism. The method requires that a user has to authenticate in order to access a device and asks for permission from a Registration Authority. The Authority, in turn, sends the user a challenge, if the answer is positive, then the user is authenticated and can access the device. Unfortunately, the proposed solution cannot prevent systems from being very vulnerable to Man in the Middle and Eavesdropping attacks. Other solutions are discussed in [29]. For each proposal, the authors examine what are the issues that are addressed and the corresponding limitations.

4. Virtual environments

Modeling and simulation techniques are essential engineering tools allowing human beings to study, analyze, understand and predict the behavior of often complex real phenomena. It is of critical importance the ability to achieve suitable mathematical models which: are accurate enough to describe the entities under investigation, are computer executable and abstract away from superfluous details. By composing models of different entities it is possible to design new complex systems which, once implemented, will interact with the real world. Simulation allows to take important decisions at design time, e.g. on the basis of the results of what-if analysis coming from the playing of different scenarios.

In the literature, two are the main categories of simulation applications that have been identified [30]: *analytic simulations* and *virtual environments*. The first includes *traditional* applications of computer simulations whose main goal is to achieve quantitative evaluations about what is being simulated and which during the execution include little or no interaction with the real world (human beings and/or physical devices). Analytic simulations are run *as fast as possible* and must supply reproducible results. Virtual Environments (VE), are systems able to simulate highly realistic environments with which people, physical devices and other systems may interact. As a consequence, a fundamental requirement for a VE is that its state must evolve at the same pace as it would in the real world so that external entities can perceive realistic feedback to their interactions with the VE. Virtual Environments are used in many areas including: military, medical, educational, emergency management and gaming. During last few years VE started to be used in the field of cybersecurity [11,31]

VEs can reuse most of the techniques that have first been devised for analytic simulations, e.g. discrete-event [30] and agent-based [9] simulations, by synchronizing the simulated time with the wallclock time.

Agents are entities able to reproduce complex human or system behavior into a scenario (real or simulated). They can operate without human interaction and perceive the environment around them. The behavior of the agents can be defined by a finite state machine [32], a Petri net or can be established by equipping the agent whit artificial intelligence algorithms [33]. Agents generally are able to cooperate and coordinate with each other in order to achieve a common goal. As will be shown in the following sections, agents are used to animate IoT scenarios by playing the roles of legal/malicious entities.

In the last few years, high-performance hardware virtualization [34,35] technologies allowed to realize complex computer networks whose nodes are virtual machines, each executing its own OS, applications and services. These technologies in turn allowed the development and diffusion of software defined networking [36] (SDN) which is currently exploited by cloud services vendors like Amazon and Google. All of this has been possible because software and protocols implementations are mathematical objects [37] and then they can be used as models of themselves.

The combined use of hardware virtualization, agent-based simulation and real devices (e.g. IoT devices) allows the realization of VEs that are suitable for the assessment of complex infrastructures in the field of Information and Communication

Technology (ICT) and, in particular, cybersecurity related aspects. Entire ICT infrastructures or relevant parts can be deployed in such a VE along with agents running on suitable simulation engines deployed on some VMs.

In this area some critical infrastructures like banking systems now have a high degree of dependence on ICT. This bound carries with it substantial advantages, e.g. automation of processes, but, on the other hand, introduces problems including security vulnerabilities. These vulnerabilities arise for several reasons, such as poor code quality. Unfortunately, many of these vulnerabilities are difficult to find, due to the systems complexity.

Different approaches have been studied in order to identify the vulnerabilities, including penetration testing. For example, a malware could be injected in a set of interconnected nodes, in order to study its propagation within the network and possible mutations. Unfortunately, this approach presents considerable risks, due to the unpredictability of the behavior of the test, which could lead to inconsistent and perhaps irreversible states of the system. A similar problem can be observed when studying the resilience of critical systems, another important research topic [38].

Researchers can rely on emulation for such analysis, especially approaches based on Emulab software [39]. Emulab is a network testbed, in which a great variety of experimental environments can be reproduced, that enable the development, testing and assessment of complex systems. In this way, the exposure of the real system to high loads and extreme conditions is avoided.

However emulators suffer from several shortcomings. The agent paradigm is not applicable, therefore the use of these software components must be reproduced by means of human intervention. It is not possible to create a distributed environment, unless the installation of a number of emulators equal to the number of nodes of the system to be analyzed is performed. Finally, one of the major difficulties in the use of emulators is due to the need of obtaining specific software or hardware components for the system to study.

For the reasons exposed above, emulators could not be considered a suitable solution for those experimental settings, that require agents or distributed configurations.

Simulation based on virtual environments is a more effective approach especially when used in conjunction with agent-based and hardware virtualization technologies that allow to abstract physical resources and specific software components.

The next subsection summarizes the main features of SmallWorld [11], a state-of-the-art virtual environment platform purposely designed for security assessment and education activities in the field of cybersecurity.

4.1. SmallWorld

SmallWorld is a software platform that has been devised in order to support the assessment, teaching and learning of security-related issues in various domains [11]. SmallWorld is based on state-of-the-art virtualization and cloud technologies for reproducing in a realistic setting a hybrid environment where large distributed computer systems can be deployed and from where they can interact with real life entities (users, software and hardware). SmallWorld enables security analysts and practitioners to design and enact complex scenarios which are dynamic and reactive and where a number of autonomous software agents can be deployed. SmallWorld agents are able to reproduce the behaviors of active entities of a given scenario, e.g. human users and/or malicious applications. This allows to the software components deployed into the virtual environment to behave and interact in a very realistic way with the actual real environment. For example, by suitably crafting the agents' behavior, the scenario may evolve over time and produce unexpected and unpredictable events that are interesting to study and analyze through simulation logs.

Fig. 1 depicts the SmallWorld architecture which is composed of five layers and has been designed to be extensible and hypervisor-independent.

The *Physical layer* hosts computational, storage and networking hardware configured in a suitable way in order to offer fault tolerance, business continuity and data replication mechanisms services for proper and scalable operation of the hypervisor.

The *Abstraction layer* virtualizes and hides hardware details which can then be easily changed/improved for scalability purposes without impacting on the overall system operations. This layer hosts the virtual machine monitor and the network hypervisor, which respectively enable to define via software the virtual computational nodes, along with the above operating systems and software layers (software defined systems) and the virtual network infrastructure (software-defined networking). There are many off-the-shelf hypervisor solutions that offers these features and that can be employed in this layer. The current prototype of SmallWorld relies on Openstack [40,41], however other implementations, i.e. VirtualBox [42], OpenNebula [43], are planned.

The *Core Service Layer* hosts the main software components that implement the core SmallWorld features which are then exposed by the overlying API layer. The blocks depicted inside this layer correspond to software components which realize specific SmallWorld functionalities. The *Network Linker* communicates with the underlying network hypervisor and introduces facilities to manage the networking services (i.e. routing, switching, bandwidth shaping, firewalling, policies). The *Publisher* is responsible to install applications (e.g. vulnerable software, malware, etc.) and agents in a given scenario. The *Datastore Engine* handles information that must be stored into suitable databases on the basis of the data type. This component does not use the abstraction layer. Data kept by the datastore engine are retrieved by the *Query Engine* and used by the *Management and Control Layer* to gather and compute statistics about the platform usage, e.g. users' and agents' activities, network bandwidth usage, traffic logs and other information.

The *Agent Engine* is basically an agent based [32–44] real-time simulation engine. It performs four main functions:

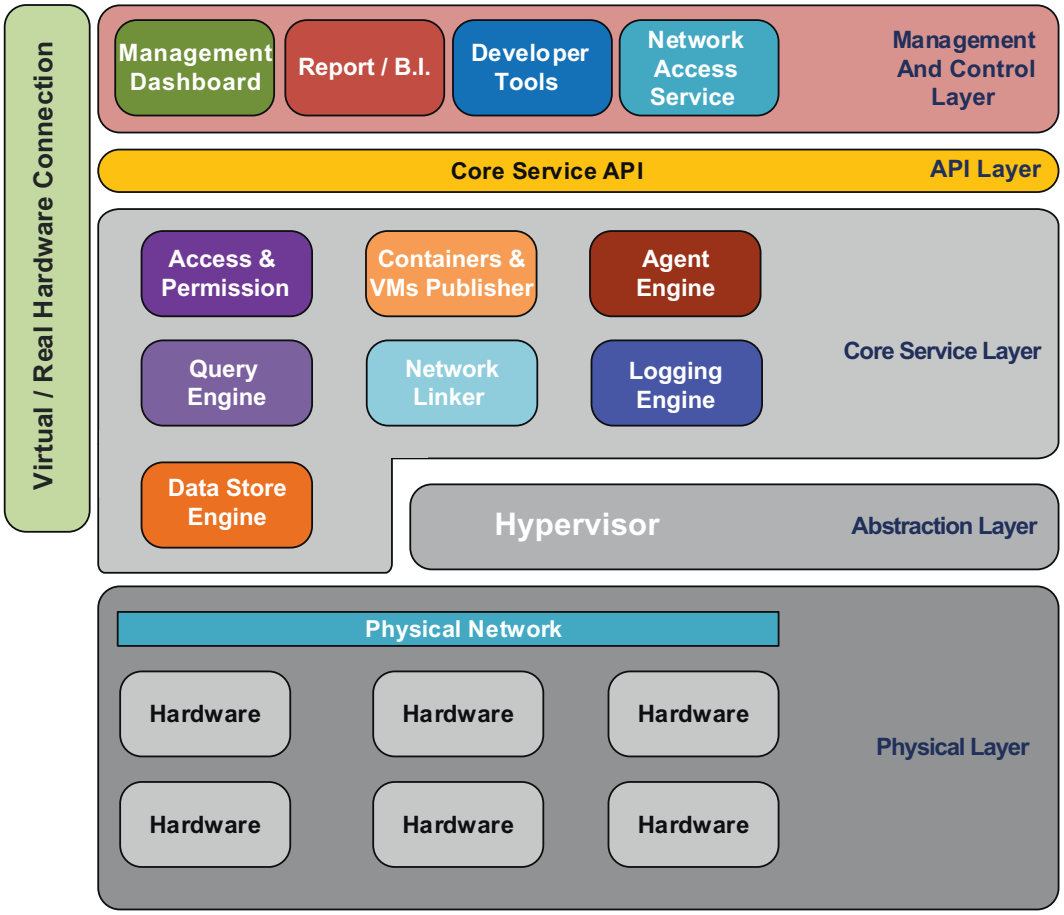

Fig. 1. SmallWorld architecture.

(i) translates Agent Behavior from a suitable agent description language (ADL) format to executable code;
(ii) provides an API for deploying and planning all simulation steps;
(iii) executes agents' behaviors in cooperation with each other providing an efficient messages delivery system;
(iv) exposes an interface to extract efficiently simulation logs.

A *Controller* entity permits to add *worker nodes* to the simulation each of which handles the execution of a little cluster of agents. The Controller orchestrates the behaviors of worker nodes.

The *API Layer* introduces a platform independent API which exposes the SmallWorld interface. This API is used for the implementation of the applications of the Management and Control Layer and it is fundamental for allowing the design and the development of reusable SmallWorld scenarios independently from the software technologies used in the underlying layers. The API is made available both as a Java framework and as a set of REST services.

The *Management and Control Layer* hosts a set of applications which ease the development and the management of SmallWorld scenarios and allow users and administrators to gather information about the status of the platform. In particular, the current version of SmallWorld provides the following tools. A *Dashboard*, enabling the management of scenarios, agents and virtual-machines. It also allows to display system usage and statics, set scenario parameters, handle user access and account management. A *Report* tool, which provides statistical data about the running scenarios. A set of *Development Tools* which include an agent development application and a scenario design tool.

SmallWorld can be exploited in various contexts and can be adapted to the specific available computational resources. Indeed, it provides different kinds of access and two type of installation: in site or in cloud. For example, an enterprise that has to deal with a large amount of data (e.g. VM images, system logs, etc.) into SmallWorld or does not want to expose private data and can afford a suitable hardware investment may opt for a in site deployment solution. The features to deliver to the client, and the respective cost, are fully customizable thanks to the modular design of the environment. On the other hand, SmallWorld can also be deployed on a cloud environment and made available as a service. This last solution allows the user to have immediate access to SmallWorld avoiding hardware investment and configuration efforts.

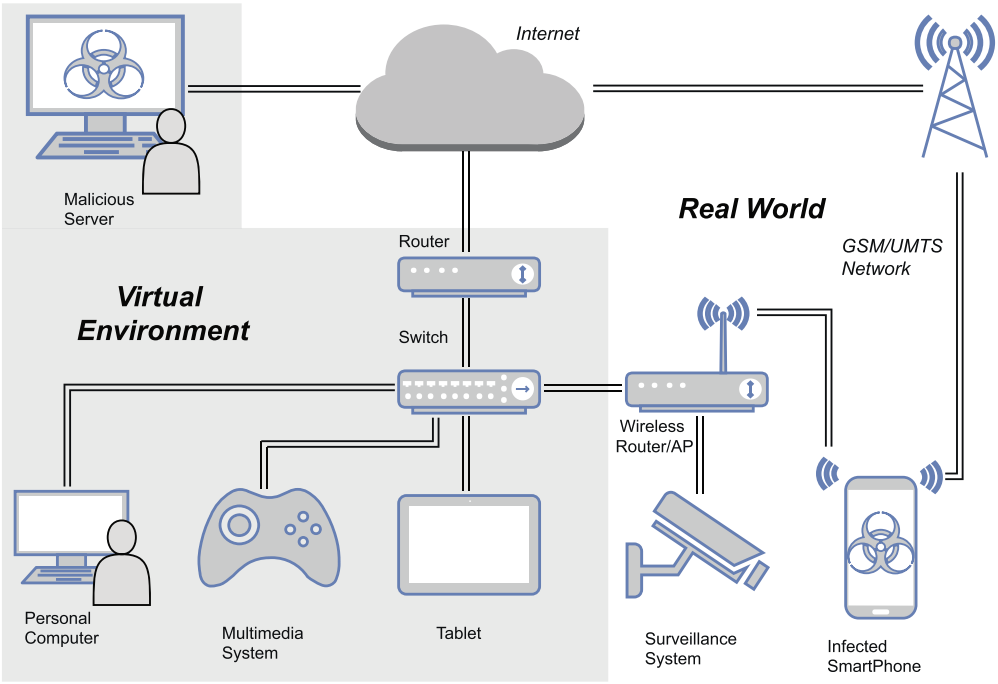

Fig. 2. A typical insecure smart home scenario.

5. Case study

This section illustrates three IoT scenarios that have been employed to investigate about the exploration and exploitation of common smart objects vulnerabilities. The scenarios were built by using the features of SmallWorld. The combined use of real devices interacting with a virtual environment allowed to analyze these IoT scenarios, assess their cybersecurity issues and conduct a suitable risk evaluation.

In particular, three variants of the same scenario were considered for studying the exposure to data leakage attacks and evaluate the effectiveness of two potential solutions.

The basic scenario is depicted in Fig. 2 and it is typical of a smart home setting. The SmallWorld virtual environment hosts some nodes that interact with the real world outside through three distinct interfaces, two of which are connected to the Internet, and one to the local network of the smart home. A *malicious* node runs in a virtual machine and it is connected to the Internet through the first interface. Albeit it is located inside the virtual environment, it has no direct access to the other virtualized nodes which are connected to a trunk of the smart home LAN. The other nodes running in VMs are: a Personal Computer intended to be used for typical users's on-line activities (e.g. browsing web-pages) and to access the video surveillance system; a multimedia system and a tablet. The tablet is emulated within the virtual environment through the use of a virtualized Android OS. These three nodes can access the Internet, through a switch which is in turn connected to a router acting as a gateway. They can also access the smart home LAN. For the purposes of the experiments, the connection to the switch is modeled as a cable connection (in a real setting it would have been a wireless connection), however this does not represent a limitation for the sake of the type of security assessment for which the scenario has been devised. Reproducing the behavior of a wirelesses connection would have been useful only in the case of evaluation of attacks such as eavesdropping or sniffing on the physical channel.

The real devices involved in the scenario are: a smart surveillance camera and an *infected* Android smartphone. The camera is directly connected to the home LAN. The smartphone can exploit two connections: one to the LAN through a WiFi access point and one to the Internet through a mobile network subscription (e.g. GSM/UMTS/LTE).

Finally, in order to animate the virtual portion of the scenario, two agents have been injected inside it: one in the VM running the Personal Computer and another in the malicious node. The former is in charge of browsing web pages and accessing the video surveillance system, the latter communicates with the infected devices sending them commands to accomplish and receiving the stolen information. The behaviors of these two agents were specified by means of state machines described in the SmallWorld agent description language. These state machines were designed and crafted on the basis of a domain expert's knowledge. In general the specification of agents behavior come from a preliminary analysis of the entities they simulate. This holds in particular for malware applications that have to be first captured and then reverse engineered [45] or their behavior inferred performing process mining activities on the logs of infected systems.

5.1. Attacking the video surveillance system

Smartphones are among the most common IoT devices and, because of their characteristics and features, represent a suitable attack vector. Such kind of devices interact everyday with different environments and establish connections not only with the home area network but also with dangerous access points as may be found in public networks. This situation, in some ways, mirrors the evolution of security on other platforms like the desktop PC, where, early attacks focused on the network layer and then migrated to the OS. An infected smartphone, which have access to a LAN of a smart home, can be easily used as the entry point to launch an attack, take control of other vulnerable IoT devices and perform malicious activities. There are many publicly available exploits for both iOS [46] and Android [47,48] devices. It is possible to unlock both Apple iPhones, by means of so called *Jailbreaks* [49], and Android smartphones, by suitable *rooting* procedures [50], and then to install vulnerable software on them. In this work, Android has been chosen because it is easier to configure the OS components due to its open-source nature and to the size of the developer community.

In the proposed scenario the remote attacker exploits the Android Stagefright Integer Overflow vulnerability (described in [51]) in order to execute remote commands. To conduct a successful attack he adds to the payload of a multimedia message the binary code of a *backdoor* [52,53]. The malicious code is sent to the victim by email, MMS, any other kind of instant messaging application or just as a link to a web page. Once the code is downloaded, it is executed as a background `telnet` service listening on port 1035 which performs tasks intended to steal information stored on the phone without the user being aware of it. The attacker remotely controls the malicious application by connecting to the backdoor port through which he can access a command shell prompt.

In the proposed scenario, when the infected smartphone connects to the home network, it starts a network scan in order to find the other IoT devices, gathering information like device model and firmware version. Such information are sent to a malicious Command and Control (c&c) server, which processes them in order to find exploitable vulnerabilities. The target of this experiment will be the Video Surveillance System, with the intent to retrieve sensitive information or to take control of it.

As the attack occurs in a LAN network, the attacker will instruct the compromised device to send spoofed Address Resolution Protocol (ARP) messages. The aim of this first phase of the attack is to associate the smartphone MAC address with the IP address of the default gateway, causing any traffic on the LAN to be sent to the attacker. Now, the attacker is able to inspect the packets and gather information, while forwarding the traffic to the actual default gateway to avoid discovery.

When the software agent placed on the personal computer try to access to the surveillance system through the web interface, it sends the credentials over the network without https encryption and the bad gateway can easily steal the credentials.

At this point the attacker, acting through the compromised device, can access the surveillance system and edit its configuration in order to make it accessible from the Internet. Considering that it is very common to find a telnet server running on such systems, often based on GNU/Linux distributions, the attacker can also login it with the stolen credentials and spawn a backdoor exploiting the ever-present `netcat` service [54].

Playing this scenario has shown how the attacker might gain sensitive information hurting the victim's privacy by means of which he can carry out criminal activities such as a blackmail. Moreover, having the control of the surveillance system, the attacker could study the victim's habits, understand when he is away from home and lead a successful robbery by turning off the home surveillance system.

5.2. Securing the smart home

The previous subsection has shown, thanks to the combined use of a virtual environment, software agents and real devices, how it is possible to reproduce a typical smart home scenario. This allowed to play inside it a real attack and to identify the security holes which are mainly due to the malware ability to access the home LAN and, as a consequence, all the connected devices.

A first step towards securing the home network consists in the installation of a suitable configured firewall as depicted in Fig. 3(a). A simple provision consists in disallowing connections to the LAN which are initiated from the external. In this case the malware is confined inside the local network. It can perform the network scan as before but the Command and Control server cannot connect to the backdoor.

However, the smartphone does not only operates within the delimited zone. It can also access to the Internet through the mobile (GSM/UMTS/LTE) network. When the device is connected in such a way, the firewall is bypassed and the malicious application can send data to the malicious server in order to receive the commands that it will subsequently execute inside the LAN. This way, the attack continues as before with the difference that the attacker sends commands to and receives data from the smartphone when the device is connected to the mobile network. In addition, the malware application must enable a rule on the firewall in order to allow the attacker to exploit the backdoor. The experiments conducted showed that this countermeasure is not adequate to properly address this attack.

A third scenario has been considered for evaluating a suitable countermeasure to the described attack. The following steps can be considered as common best practices [55] to achieve a basilar home-protection:

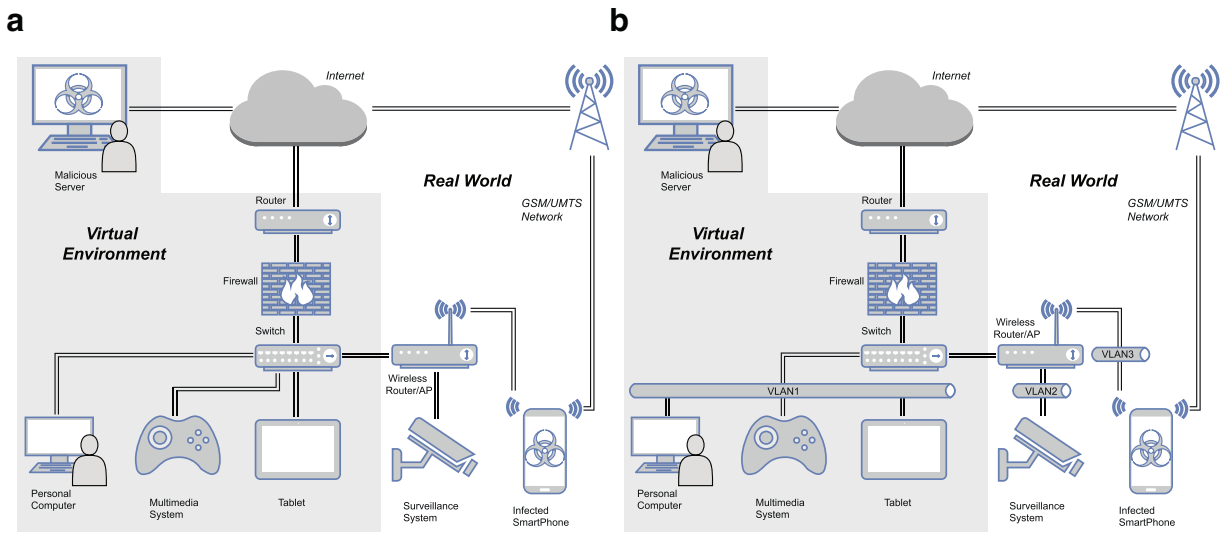

Fig. 3. Scenario configuration with: (a) firewall, (b) separate VLANs.

- identify which devices need to be protected;
- group devices into logical groups;
- identify critical and not-critical groups;
- isolate each group in a separate sub-network and monitor activities that occur among them.

Each of the considered devices should be evaluated independently in order to find which one is part of critical assets. Being able to make such a separation, in many cases, is not an easy task. Indeed, the identification of critical devices is not only linked to critical assets to which they have direct access, but also to the logical dependencies that exist among them. Logical dependencies have a key role in this task, and if they are not suitably handled they can easily become *Trojan horses* through which access to critical assets can be gained. Based on such type of analysis, a logical separation among the devices, ensuring a suitable level of security, can be derived.

Following these guidelines it is easy to identify the surveillance system as the most critical device. As a consequence, it must necessarily be placed in a separate group and must have the least possible interaction with other devices. Another group is represented by *non mobile* devices, which in our scenario are confined exclusively within the home, i.e. the multimedia system, the tablet and the PC. Devices that can connect to other networks, such as the smartphone in the considered setting, must be part of a separate group because they are potential attack vectors. Another group, not present in this scenario, could be one reserved to host devices.

To isolate the logical groups identified in the previous phase, the home network is divided into three sub-networks, by resorting to the use of Virtual LANs, corresponding to the above-described devices groups. Using this configuration, the smartphone, which is the main attack vector, is isolated and cannot perform scanning in the whole network in order to discover the devices inside the smart home.

6. Conclusions

The pervasive diffusion of smart devices is going to change many aspects of our daily lives and also the way how most business activities will be accomplished in the next future. Just think how the smartphone already changed the way people communicate, make their appointments and plan their travels. The next step will be to deal with the consequences of the worldwide spread of IoT devices. Because human beings are going to delegate many (critical) activities to smart devices, it is of utmost importance to suitably cope with cybersecurity issues threatening IoT. Most part of the IoT related literature is about enabling technologies and applications. Security problems are overlooked even if recent studies claim that they will have an enormous economic impact in the next years. This paper shows how simulation technologies can be effectively used for the assessment of cybersecurity scenarios involving IoT settings. In particular, the combined use of novel virtual environments, able to exploit state-of-the-art hardware virtualization technologies and cloud computing, agent-based simulation and real devices allow to design and evaluate, in a controlled way, IoT technologies (applications, protocols, device prototypes) and related security issues before releasing them in production. The effectiveness of the proposed approach is demonstrated through a case study regarding a typical smart home setting which is evaluated by means of the SmallWorld platform.

Acknowledgments

This work has been partially supported by the "National Operative Programme for Research and Competitiveness" 2007-2013, Technological District on Cyber Security (PON03PE_00032_2_02), funded by the Italian Ministry of Education, University and Research, and the Italian Ministry of Economic Development.

References

[1] F. Mattern, C. Floerkemeier, From the Internet of computers to the Internet of things, in: From Active Data Management to Event-Based Systems and More, in: Vol. 6462 of LNCS, Springer, 2010, pp. 242–259, doi:10.1007/978-3-642-17226-7_15.
[2] Fleisch(2010) E. Fleisch, What is the internet of things? An economic perspective, white paper WP-BIZAPP-053, auto-ID labs, 2010.
[3] L. Atzori, A. Iera, G. Morabito, The internet of things: a survey, Comput. Netw. 54 (15) (2010) 2787–2805, doi:10.1016/j.comnet.2010.05.010.
[4] Gartner. Gartner says 6.4 billion connected "thing" will be in use in 2016, up 30 percent from 2015, http://www.gartner.com/newsroom/id/3165317 (10 Nov. 2015).
[5] The digital universe of opportunities: Rich data and the increasing value of the Internet of Things, https://www.emc.com/collateral/analyst-reports/idc-digital-universe-2014.pdf (April 2014).
[6] Gartner, Gartner says by 2020, more than half of major new business processes and systems will incorporate some element of the Internet of Things, http://www.gartner.com/newsroom/id/3185623 (14 Jan. 2016).
[7] B. Anderson, S. Barsamian, D. Childs, J.D.a.J.M. Forsythe, B. Gorenc, A. Gunn, A. Hoole, H. Miller, S.S. Muthurajan, Y.T. ONeil, J. Park, O. Petrovsky, B. Raz, N. Shah, V. Svajcer, K. Tietjen, J. Timpe, Cyber Risk Report 2016, Tech. rep., Hewlett Packard Enterprise, 2016.
[8] R. Neisse, G. Steri, I.N. Fovino, G. Baldini, Seckit: a model-based security toolkit for the internet of things, Comput. Secur. 54 (2015) 60–76, doi:10.1016/j.cose.2015.06.002.
[9] F. Cicirelli, A. Furfaro, L. Nigro, Exploiting agents for modelling and simulation of coverage control protocols in large sensor networks, J. Syst. Software 80 (11) (2007) 1817–1832, doi:10.1016/j.jss.2007.02.015.
[10] F. Cicirelli, A. Furfaro, A. Giordano, L. Nigro, HLA_ACTOR_REPAST: an approach to distributing RePast models for high-performance simulations, Simul. Modell. Pract. Theory 19 (1) (2011) 283–300, doi:10.1016/j.simpat.2010.06.013.
[11] A. Furfaro, A. Piccolo, D. Saccà, A. Parise, A virtual environment for the enactment of realistic cyber security scenarios, in: Proceedings of 2nd IEEE International Conference on Cloud Computing Technologies and Applications (CloudTech 2016), Marrakesh, Morocco, 2016.
[12] M. Ge, D.S. Kim, A framework for modeling and assessing security of the Internet of Things, in: 2015 IEEE 21st International Conference on Parallel and Distributed Systems (ICPADS), Institute of Electrical & Electronics Engineers (IEEE), 2015, pp. 776–781, doi:10.1109/icpads.2015.102.
[13] K.C. Park, D.-H. Shin, Security assessment framework for IoT service, Telecommun. Syst. (2016) 1–17, doi:10.1007/s11235-016-0168-0.
[14] W. Leister, S. Poslad, M. Hamdi, H. Abie, A. Torjusen, An evaluation framework for adaptive security for the IoT in eHealth, Int. J. Adv. Secur. 7 (3–4) (2014) 93–109.
[15] W. Aman, E. Snekkenes, Managing security trade-offs in the internet of things using adaptive security, in: 2015 10th International Conference for Internet Technology and Secured Transactions (ICITST), IEEE, 2015, pp. 362–368, doi:10.1109/icitst.2015.7412122.
[16] H. Ghani, A. Khelil, N. Suri, G. Csertán, L. Gönczy, G. Urbanics, J. Clarke, Assessing the security of internet connected critical infrastructures (the CoMiFin project approach), in: Proceedings of the Workshop on Security of the Internet of Things, 2010.
[17] C. Bekara, Security issues and challenges for the IoT-based smart grid, Procedia Comput. Sci. 34 (2014) 532–537, doi:10.1016/j.procs.2014.07.064.
[18] D. Halperin, T.S. Heydt-Benjamin, B. Ransford, S.S. Clark, B. Defend, W. Morgan, K. Fu, T. Kohno, W.H. Maisel, Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses, in: 2008 IEEE Symposium on Security and Privacy (SP 2008), IEEE, 2008, pp. 129–142, doi:10.1109/sp.2008.31.
[19] S. Babar, P. Mahalle, A. Stango, N. Prasad, R. Prasad, Proposed security model and threat taxonomy for the internet of things (IoT), in: Recent Trends in Network Security and Applications, Springer, 2010, pp. 420–429, doi:10.1007/978-3-642-14478-3_42.
[20] L.A. Berk, S.F. Oehninger, The risk of insuring supply chains from cyber risk, Law360 (Jun. 29 2015). http://www.law360.com/articles/668004/the-risk-of-insuring-supply-chains-from-cyber-risk.
[21] A.W. Atamli, A. Martin, Threat-based security analysis for the internet of things, in: 2014 International Workshop on Secure Internet of Things, IEEE, 2014, pp. 35–43, doi:10.1109/siot.2014.10.
[22] C. Miller, C. Valasek, Remote exploitation of an unaltered passenger vehicle, Black Hat USA, Mandalay Bay, Las Vegas, NV, USA, 2015.
[23] Somerset Recon, Hello Barbie security: Part 2 - analysis, http://www.somersetrecon.com/blog/2016/1/21/hello-barbie-security-part-2-analysis (Jan. 2016).
[24] B. Parno, A. Perrig, V. Gligor, Distributed detection of node replication attacks in sensor networks, in: 2005 IEEE Symposium on Security and Privacy (S&P'05), IEEE, 2005, pp. 49–63, doi:10.1109/sp.2005.8.
[25] M. Asadi, C. Zimmerman, A. Agah, A game-theoretic approach to security and power conservation in wireless sensor networks, in: International Journal of Network Security, volume 15, 2013, pp. 50–58.
[26] S. Yan-qiang, W. Xiao-dong, Jamming attacks and countermeasures in wireless sensor networks, in: From Principle to Practice, IGI Global, 2010, pp. 34–352.
[27] Symantec, An internet of things reference architecture, White paper 2016). https://www.symantec.com/content/dam/symantec/docs/white-papers/iot-security-reference-architecture-en.pdf.
[28] J. Liu, Y. Xiao, C.P. Chen, Authentication and access control in the internet of things, in: 2012 32nd International Conference on Distributed Computing Systems Workshops, Institute of Electrical & Electronics Engineers, (IEEE), 2012, pp. 588–592, doi:10.1109/icdcsw.2012.23.
[29] S.A. Kumar, T. Vealey, H. Srivastava, Security in Internet of things: challenges, solutions and future directions, in: 49th Hawaii International Conference on System Sciences (HICSS), Institute of Electrical & Electronics Engineers (IEEE), 2016, pp. 772–5781, doi:10.1109/hicss.2016.714.
[30] R.M. Fujimoto, Parallel and Distributed Simulation Systems, John Wiley & Sons, Inc., 2000.
[31] S. Schwab, B. Wilson, C. Ko, A. Hussain, SEER: a security experimentation environment for DETER, USENIX Association, 2007. Proceedings of the DETER Community Workshop on Cyber Security Experimentation and Test on DETER Community Workshop on Cyber Security Experimentation and Test 2007
[32] F. Cicirelli, A. Furfaro, L. Nigro, Modelling and simulation of complex manufacturing systems using statechart-based actors, Simul. Modell. Pract. Theory 19 (2) (2011) 685–703, doi:10.1016/j.simpat.2010.10.010.
[33] M. Woolridge, M.J. Wooldridge, Introduction to Multiagent Systems, John Wiley & Sons, Inc., New York, NY, USA, 2001.
[34] S. Hand, High-performance virtualization: are we done? Commun. ACM 59 (2015) 107, doi:10.1145/2845910.
[35] N. Amit, A. Gordon, N. Har'El, M. Ben-Yehuda, A. Landau, A. Schuster, D. Tsafrir, Bare-metal performance for virtual machines with exitless interrupts, Commun. ACM 59 (1) (2015) 108–116, doi:10.1145/2845648.
[36] D. Kreutz, F.M.V. Ramos, P.E. Verissimo, C.E. Rothenberg, S. Azodolmolky, S. Uhlig, Software-defined networking: a comprehensive survey, Proc. IEEE 103 (1) (2015) 14–76, doi:10.1109/jproc.2014.2371999.
[37] C. Ghezzi, M. Jazayeri, D. Mandrioli, Fundamentals of Software Engineering, 2nd edition, Pearson, 2002.
[38] B. Genge, C. Siaterlis, Developing cyber-physical experimental capabilities for the security analysis of the future smart grid, in: 2011 2nd IEEE PES International Conference and Exhibition on Innovative Smart Grid Technologies, Institute of Electrical & Electronics Engineers (IEEE), Manchester, 2011, pp. 1–7, doi:10.1109/isgteurope.2011.6162766.

[39] C. Siaterlis, A.P. Garcia, B. Genge, On the use of emulab testbeds for scientifically rigorous experiments, IEEE Commun. Surv. Tutorials 15 (2) (2013) 929–942, doi:10.1109/surv.2012.0601112.00185.
[40] G. Aubuchon, K. Cacciatore, M. Fainberg, C. Hoge. Expediting digital workflow with openstack, White paper (2015). https://www.openstack.org/assets/pdf-downloads/OpenStack-Workflow-White-Paper-Letter-Final.pdf.
[41] O. Sefraoui, M. Aissaoui, M. Eleuldj, Openstack: toward an open-source solution for cloud computing, Int. J. Comput. Appl. 55 (3) (2012) 38–42, doi:10.5120/8738-2991.
[42] G. King, 2012, Oracle VM 3: Building a demo environment using Oracle VM VirtualBox, http://www.oracle.com/technetwork/server-storage/vm/ovm3-demo-vbox-1680215.pdf.
[43] D. Milojičić, I.M. Llorente, R.S. Montero, OpenNebula: a cloud management tool, IEEE Internet Comput. 15 (2) (2011) 11–14, doi:10.1109/mic.2011.44.
[44] R. Axelrod. The complexity of cooperation: Agent-based models of competition and collaboration, Walter de Gruyter GmbH, 1997. http://dx.doi.org/10.1515/9781400822300.
[45] M. Bombardieri, S. Castano, F. Curcio, A. Furfaro, H.D. Karatza, Honeypot-powered malware reverse engineering, in: 2016 IEEE International Conference on Cloud Engineering Workshop (IC2EW), Institute of Electrical and Electronics Engineers (IEEE), 2016, doi:10.1109/ic2ew.2016.16.
[46] CVE Details, Apple, iphone os: Security vulnerabilities. http://www.cvedetails.com/vulnerability-list/vendor_id-49/product_id-15556/Apple-Iphone-Os.html.
[47] Androidvurnelrabilities.org, http://androidvulnerabilities.org/all.
[48] D. Vecchiato, M. Vieira, E. Martins, The perils of android security configuration, Computer 49 (6) (2016) 15–21, doi:10.1109/mc.2016.184.
[49] iOS Jailbreak, http://www.ios9cydia.com/.
[50] Y. Shao, X. Luo, C. Qian, RootGuard: protecting rooted android phones, Computer 47 (6) (2014) 32–40, doi:10.1109/mc.2014.163.
[51] J. Drake, Stagefright: Scary Code in the Heart of Android, Las Vegas, NV, USA, 2015. https://www.blackhat.com/docs/us-15/materials/us-15-Drake-Stagefright-Scary-Code-In-The-Heart-Of-Android.pdf. Black Hat USA, Mandalay Bay.
[52] I. Arce, The shellcode generation, IEEE Secur. Privacy Mag. 2 (5) (2004) 72–76, doi:10.1109/msp.2004.87.
[53] S. Padilla. Android-Telnetd (port 1035) with parameters shellcode (248 bytes), https://www.exploit-db.com/exploits/38194/.
[54] The GNU Netcat project, http://netcat.sourceforge.net/.
[55] E.D. Knapp, J.T. Langill, Industrial Network Security: Securing Critical Infrastructure Networks for Smart Grid, SCADA, and Other Industrial Control Systems, Syngress Publishing, 2011.