Intrusion detection in Cloud Internet of Things Environment

Mohammed REBBAH
University of Mustapha Stambouli, Mascara, ALGERIA
LRSBG Laboratory, TVIM Group
Rebbah_med@yahoo.fr, mohammed.rebbah@univ-mascara.dz

Dhiaa El Hak REBBAH
University of Mustapha Stambouli, Mascara, ALGERIA
dhiaarebbah@gmail.com

Omar SMAIL
University of Mustapha Stambouli, Mascara, ALGERIA
smailomar@gmail.com

***Abstract*— The insufficient amount of storage, the slow speed of data and request manipulation and the guarantee of the whole system's security. In order to resolve such challenges, the only solution was to integrate the Cloud to the Internet of Things; this gave us the opportunity to achieve an Internet of Things project but not to maintain the security of the whole IoT environment. In this paper, we tackle the problem of the intrusion in this environment. We propose, IoTSecurity, an intrusion detection system based on Signature-based approach. IoTSecurity calculate Temporary and spatial profile of each client based on the data of its request. Temporary and spatial profile are studied and tested to evaluate the proposed model.**

Keywords*— *Internet of Things; CloudIoT; Intrusion Detection System; Temporal Profile; Spatial profile.

I. INTRODUCTION

The Internet of Things is going to be the next evolution in the era of Internet Technology, where the main concept about this new paradigm is to allow connectivity anywhere and at anytime by allowing objects to abstract information and share it without human interfering [1]. By using the RFID (Radio Frequency Identification) [1], we can make objects actually connect with each other and share information in order to make life a better place. The main problems with the Internet of Things project are the objects themselves: for example, the lack of storage space for each individual object while we have billions of objects, and the processing performance of each object, the maintaining of a durable communication between the objects, the security as well as the confidentiality of them and also the information collected by each one. Therefore, CloudIoT is a fusion between two different aspects: Cloud Computing and the Internet of Things [2]. Cloud provides the unlimited capacity of storage and the highest possible speed and performance of execution. It also provides the durably required communication. The only problem is how to guarantee the security and the confidentiality of both the information and the whole Internet of Things system. We are, therefore, led to a different aspect consisting in the IDS (Intrusion Detection System) [3]: which is probably the most convincing solution to maintain such complicated system secured. So, an IDS must be installed inside the CloudIoT to keep full control on the information from one side and the requests/responses transaction between the CloudIoT and the clients from the other side. IDS can control the circulation of information which means it may influence the confidentiality of clients. This explains why only few IDS were made for a CloudIoT project. As a solution to such problem, we propose an IDS that can detect intrusion based on the Requests details, and User's profile using the data provided by the execution of requested on the CloudIoT servers without checking the information circulating in the system, but by checking the user's profile, it detects intrusion by a suspected behavior of each client.

Our contributions in this paper are:

- We detect the intrusion without any document analysis and study the data provided by the client that has submitted a request to the cloud server
- Our IDS is based on the temporal profile and spatial profile calculated for each client
- We preserve the client privacy

The remainder of this paper is structured as follows: the second section presents a related work about IDS in IoT and Cloud Computing. We present, in section 3, IoTSecurity, the proposed IDS in CloudIoT. Section 4 provides the results of our experiments and we finish, this paper, by a conclusion and perspectives.

II. RELATED WORK

Basically, because the technology of the Internet of Things is one of the recent results of the Internet Technology era, the application must be infinite in different areas in our world where the main objective is to make Objects connect with each other and make decisions without human interference. So the main goal is to accomplish this connectivity. However, achieving that goal has never been easy: going through the performance and connectivity issues and then using the CloudIoT solution. A project of the Internet of Things can be finally accomplished, but the problem is to get all the amount of information circulating in this huge Network under control by installing an Internet of Things IDS. So far, only two IDS compatible are to be installed in an IoT environment: CloudAV [5] and CloudEyes [4]. There are different IDS such

978-1-5386-3269-7/17/$31.00 ©2017 IEEE

as Snort [3]. Besides installing such an Intrusion Detection System in a huge Network can put the IDS out of control. Each IDS has its own strategy of detecting intrusions in a reliable way, but it differs in the way of scanning information in order to detect that intrusion.

a) Internet of Things:

The contemporary era of the future internet has already been deployed, in the form of innovative Internet of Things (IoT) infrastructures, in different application domains that range from smart health and smart buildings to smart cities [9]. The term Internet of Things was first coined by Kevin Ashton in 1999 in the context of supply chain management. However, in the past decade, the definition has been more inclusive covering wide range of applications like healthcare, utilities, transport, etc. [6].

The technology of IoT was considered as an upgraded version of the Internet itself, because since 1969 (The birthday of the Internet) we did not get any sort of evolution and basically the Internet needed to be upgraded. First, the Internet was the Network of computers where the main idea is to allow connectivity between computers, and then the Internet started moving to connecting smart phones along side with computers to create the vast Network we know currently. Based on studies by Cisco IBSG made in 2015 "there are almost 25 billion devices connected to the Internet, and it is supposed to rise to reach nearly the roof of 50 billion devices connected by 2020" [7]. IoT is a concept that gathers all sorts of different applications based on the convergence of smart objects and the Internet, establishing an integration between the physical and the cyber worlds. These applications may range from a simple appliance for a smart home to a sophisticated equipment for an industrial plant. Although IoT applications have very different objectives, they share some common characteristics.

Generally speaking, IoT operations include three distinct phases: collection phase, transmission phase, and processing, management and utilization phase [10]. The numbers are always rising, so the idea to accomplish an Internet of Things project is first to link every mobile and fixed devices around the globe with each other and with the Cloud servers, and then let the Information gathered by each of these devices be ubiquitous without human interference. This can only be reached by using the sensors installed on every single device, and also by providing a connectivity that allows the Information shared between these devices be as Objects or Things, forming the so-called the Internet of Things. After assembling and connecting all the mobile and fixed devices around the globe, we move on to the next step into achieving a completed Internet of Things project by getting the non-smart objects such as traffic lights connect in order to gather information about the traffic situation. Moreover, medical devices can collect data about patients and share it directly to the doctors or the medical staff. After completing this procedure, we will an Internet of Things project.

b) Intrusion Detection System: The IDS is made to be installed in a system for a main goal which is detecting intrusions obviously, it could be a hardware IDS or only software [3].

Intrusion detection techniques are classified into four categories depending upon the detection mechanism used in the system: anomaly-based, signature-based, specification based and hybrid [11]. In order to detect intrusion in a CloudIoT environment the required IDS is mainly software, therefore only two intrusion detection systems were found till the moment, these two softwares are CloudAV [5] and CloudEyes [4]. Both of these mentioned IDS are for a Gigabits/sec rate which means a huge amount of information can be treated normally and the system should function as well. Modi et al. [12] report several intrusions that affect availability, confidentiality, and integrity of Cloud Computing. The authors summarize and classify IDSs used in Cloud into three categories: IDS technology (Host-based intrusion detection system (HIDS), Network-based intrusion detection system (NIDS), Hypervisor based intrusion detection system and Distributed intrusion detection system (DIDS)), detection technique and network positioning. They also discuss advantages and disadvantages of each proposal and identify challenges to make Cloud Computing a trusted platform for delivering IoT services. Most of the proposed intrusion detection techniques in Cloud cannot deal with recurrent attacks in this environment such as the insider attacks and attacks on the virtual machine or hypervisor. The only thing to be concerned about is: What is the Strategy followed to detect intrusion by each IDS?

b.1) CloudAV: Detecting malicious software is a complex problem. The vast, ever-increasing ecosystem of malicious software and tools presents a daunting challenge for network operators and IT administrators. [5]. Basically, CloudAV is an antivirus made essentially for Cloud Computing purposes, in order to keep the whole Cloud servers secured. CloudAV uses 12 engines: 10 traditional antivirus engines such as Avast, Kaspersky and McAfee, it also uses 2 behavioral engines (Norman Sandbox, CWSandbox). The idea of CloudAV is to detect intrusion in the information that is being transacted between the clients and the Cloud Server, but there is no confidentiality about it, because CloudAV needs to test all the files and documents and any kind of data that is circulating in the Cloud or the Internet of Things environment. This is one of the inconvenient about CloudAV, even the intrusion detection is totally accurate based on twelve of the best antivirus engines in the world and on two developed behavioral engines to detect the behavioral changes about the users which would give an idea about an intrusion detected. The only problem is that the users (clients) want the confidentiality guaranteed by the whole system, and that what CloudAV can't provide.

b.2) CloudEyes: Because of the rapid increasing of malware attacks on the Internet of Things, it is critical for resource-constrained devices to guard against potential risks, because in such environment as the Internet of Things, the objects are everywhere and it is possible for any hacker to get access to the information captured by that object. This requires to install a good IDS in such a vast environment with a huge amount of data that treats every piece of information captured by objects and also requires to keep these objects secured from outside

978-1-5386-3269-7/17/$31.00 ©2017 IEEE

interactions. CloudEyes is malware detection in an Internet of Things environment that detects intrusions based on the signature-based detection method where the purpose is to detect malwares in the data circulating in the IoT system, but also preserving the privacy which was the main motivation of creating CloudEyes. In the Cloud server, CloudEyes uses a novel signature-based detection mechanism called: "suspicious bucket cross-filtering". This gives an accurate orientation of the malicious signature fragments. For the client, CloudEyes will make a lightweight scanning just in order to detect the suspicion of file content without analyzing the whole data, but only by analyzing the digest of reversible sketch. CloudEyes was made for a purpose to protect both Cloud and Client server's privacy and reduce communication consumption. The problem with CloudEyes is that it is hard to achieve simultaneously the data privacy protection and the higher performance of security detection that CloudAV provides. CloudAV provides the highest performance of security detection ever, but the problem is it messes with the Client's privacy, but CloudEyes provides the data privacy protection but with a lower degree in security detection compared to CloudAV.

In this paper, we detect the intrusion by exploiting the data provided by the client when submitting requests to the cloud server without any document analysis.

III. PROPOSED MODEL

In order to accomplish our researches and achieve a goal of developing an Intrusion Detection System, we were required to make a CloudIoT environment that regroups the Internet of Things and the Cloud components in one vast environment. We proposed our IDS, called IoTSecurity, which exploit the data provided from the Cloud server, the Clients (Objects in the Internet of Things) and the client request.

In order to become a Client in such environment, all new objects are required to sign a contract, whether the object is mobile or fixed. The contract is signed between the Clients and the Cloud provider, when the new object wants to become a client, it must follow the sign up steps introduced by the Cloud Server. These steps contain information such as the object's identificator, the certificate of the object, and also the number of requests by each object. In the Cloud area, we exploit another amount of useful information, such as: the number of virtual machines (VM), the specification of each virtual machine, such as the amount of available MIPS (Million Instruction per Second), RAM, Storage Size and also compatible Bandwidth. We did not forget about the movement of mobile devices. So, we took care of that side as well. Each mobile object has then its own position an *X* and a *Y*. In order to detect the intrusion based on the current location of each object, we had to consider that this concerns only the mobile objects.

A. *Data source*

In our model, we have three different data sources. These three different sources are:

A) CloudClients: that contains the information about each object, such as the identificator, the type of object either a mobile or fixed one, the position of the object and the number of demanded requests. These data are modeled as *Client(Id_client, Type_client, Pos_X, Pos_Y, Number_request).*

B) CloudPlatfom: and this is the source that contains the necessary information that concerns the Cloud environment such as: the number of available virtual machines, the identificator of each virtual machine, the available MIPS, RAM, storage size and the possible bandwidth of each virtual machine. Which are modeled as *CloudPlatform(id_VM, MIPS, RAM, StorageSize, Bandwidth).*

C) Requests: this source is dedicated for the circulating requested in the CloudIoT environment, whether it's a mobile or a fixed object, the idea is to maintain control on all the requests executed in the environment, and recorded in these data in order to study a case of intrusion. These data source is modeled as *Request_client(Xd, Yd, start_time, submit_date, ExecuteTimeD, MipsD, RamD, StorageSizeD, BandwithD).*

B. *Intrusion detection system*

So, in order to detect intrusions, our IDS IoTSecurity will be installed in the Cloud Server, and it must exploit every data from the three data source mentioned above, and the other data sets such as Certificate data set that contains all the certificates of each object…etc. (See Figure 1).

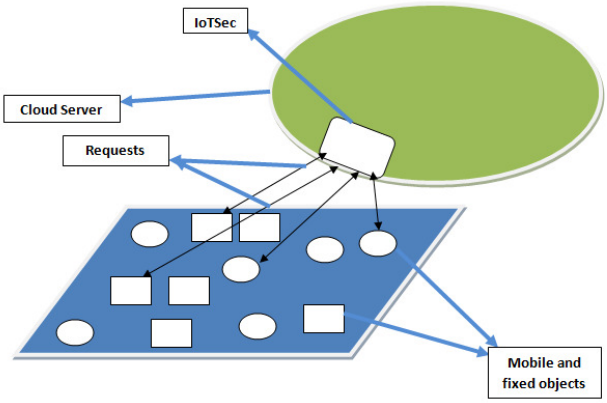

Fig. 1. The CloudIoT environment

IoTSecurity's intrusion detection method goes through three steps consecutively, and every step is necessary in order to accomplish accurate intrusion detection. First, when the objects want a request to be executed, they send the request to

the Cloud Server, IoTSecurity interferes in order to detect intrusion based on the certificate validation: IoTSecurity will compare the certificate of the object demanding the request with the certificates registered in the Certificate data set. If the certificate is not valid, IoTSecurity will detect the object as an intruder and cancel the execution of the request on demand immediately. After the complete intrusion detection based on Certificate level, IoTSecurity moves to the next step which is: detecting intrusion based on the user's profile. This requires a new database called profile that contains all the information about each object. If we have an object mobile with the identificator number 5, and based on its provided certificate which is valid, IoTSecurity will allow the execution of the requests that Mobile Device number 5 is requesting, but based on the previous 10 requests executed. Our IDS will generate a profile for this device, the profile contains some valuable information about the requests demanded by this object, and according to this amount of data, IoTSecurity will detect intrusion if the request number 11 is a bit different from that of the average of that profile. The detection based on User's profile is divided into two steps as well:

Detecting Intrusion based on temporary profile: it allows IoTSecurity to detect intrusions based on speed differentials, by calculating the average speed of that object as follows:

$$AV\_speed=Means(\frac{\sqrt{(x_2-x_1)^2+(y_2-y_1)^2}}{T_2-T_1},\dots,\frac{\sqrt{(x_n-x_{n-1})^2+(y_n-y_{n-1})^2}}{T_n-T_{n-1}}) \quad (1)$$

Where (x_n, y_n, T_n) and $(x_{n-1}, y_{n-1}, T_{n-1})$ are Xposition, Yposition, Starttime of two last requests respectively.
IoTSecurity detects an intrusion if and only if the object's speed exceeds the average speed value in an abnormal way. For example, the average speed of the mobile object number is 40km/h in a perimeter of 150 km^2. If the next request is executed in a very distant place, that will generate a new speed value exceeding the 40 km/h and will make IoTSecurity detects it as an Intrusion.

Detecting Intrusion based on spatial profile: This type of detection is the last step in IoTSecurity's detection phases. For each client, we calculate his gravity center which represents his spatial profile. We use the request data of the entire request client executed in the cloud platform. Each request is represented by Request_client(Xd, Yd, start_time, submit_date, ExecuteTimeD, MipsD, RamD, StorageSizeD, BandwithD). For each variable, we calculate its average as follows:

$$\text{Mean (Xd)} = \frac{\sum \text{Xd}}{n} \quad (2)$$

Where n is the number client request.

Based on different types of averages such as the average required MIPS, the average required RAM, the average required storage as well as Bandwidth, the gravity center of each client is calculated as follow:

$$\text{GC (Client X)} = (\text{Mean (Xd)}, \dots, \text{Mean(Bandwidth)}) \quad (3)$$

After the gravity center calculation, we determine the farthest request belonging to the client requests. This most far request is the "The most far request based on the gravity center" (See Figure 2). Which is calculated as follows:

$$\text{Max}(\frac{\sqrt{(x2-x1)^2+(y2-y1)^2}}{\sum(x,y)},\frac{\sqrt{(x2-xg)^2+(y2-yg)^2}}{\sum(x,y)}) \quad (4)$$

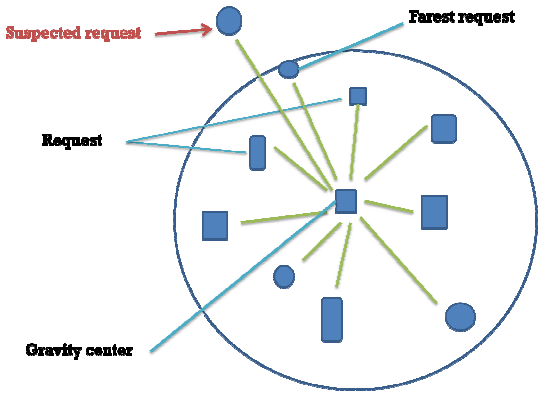

Fig. 2. Intrusion detection based on spatial profile

When a client submits a new request with all its parameters, we create a new line data from the data source of request, as *Request_client(Xd, Yd, start_time, submit_date, ExecuteTimeD, MipsD, RamD, StorageSizeD, BandwithD).* We calculate its distance from the gravity center and we compare this distance with the distance of the most Far request. If the new request is farther than the farthest one, IoTSecurity detects an intrusion based on spatial profile.
If an object is not detected on the three steps of detection, the request on demand will be executed successfully by the Cloud Server. IoTSecurity updates the profile of each client after an execution of a new request.

C. IoTSecurity architecture

Our intrusion detection system, called IoTSecurity, is installed in the cloud server and is composed of two modules request submission and request execution. Certificate database and user profile databases are generated from the data source (See Figure 3).
Request submission module: To make IoTSecurity work, the first step is setting up the Requests Database so as to create the Cloud Clients and Cloud Platform databases. On first execution, IoTSecurity gets all the data about the clients and the Cloud Server with the whole virtual machines information in order to generate the profiles.
Request execution module: After the execution of the first requests, there is another generation to a different database, such as Signatures, that contains the list of certificates of each client. Besides, the Speed database generation contains the movement of each client, the amount of RAM, Storage Size, Bandwidth and also the MIPS requested from each client.
Signatures database: This is the moment when the IDS does the first step of checking intrusions before even letting the execution of any request. IoTSecurity will check the certificate

978-1-5386-3269-7/17/$31.00 ©2017 IEEE

of each demanding Client/Object and compare it with the certificates stored in the Signatures/Certificates database. So, if there is no match, the request won't be executed at all and the object will be detected as intruder.

User Profile database: After executing the requests successfully, IoTSecurity will generate a database from the Cloud Clients Database and the Speed entitled Profiles Database will allow the IDS to detect intrusions based on abnormal requests or movements of objects. The user profile contains: the identificator of each object, the farthest position this object goes to, the most highest valuable of RAM, MIPS, Storage Size and Bandwidth that this object demanded from the Cloud Server. Based on this amount of data IoTSecurity will compare the next upcoming requested of each object. If there is any abnormal request in Virtual Machines specifications or if the position is very far from the normal position, it will, then, be detected as an intruder and the request will not be executed.

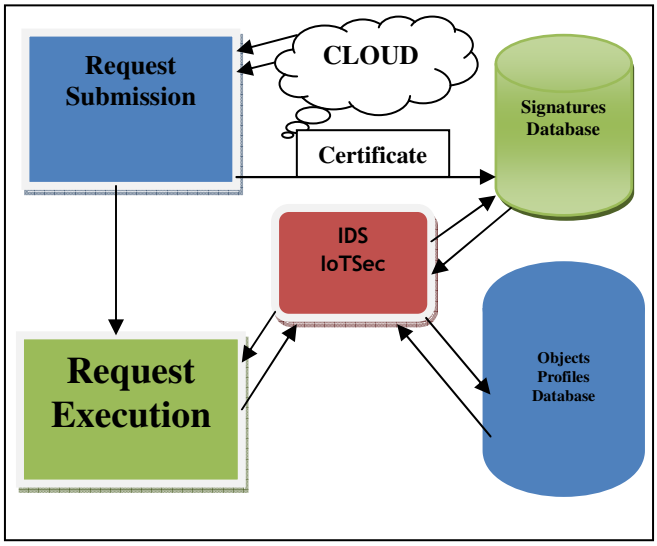

Fig. 3. IoTSecurity architecture

IV. Experiments and Results

Because of the lack of CloudIoT environments as we have mentioned earlier, the only available platform was iFogSim [8]. Consequently, we could neither manage to manipulate the information's circulation in such environment nor achieve a goal of creating an IDS and integrate it into iFogSim. So, the main idea was to develop firstly our own CloudIoT platform with all the components of the cloud computing environment, secondly the quantity of objects required both to scan and detect as intruders, thirdly our own IDS make a connection between it and the platform itself. This is basically what we have achieved with IoTSecurity. In the next step, we tested the performance of our IDS on different levels. We first generate the different databases such as Cloud Platform containing information about the virtual machines and the Cloud Servers, the Cloud Clients containing all the data about the Objects we were treating, the request database containing information about the requests specifics of each object with the Cloud Server and finally the Profile and Certificate databases. We tested the application by adding some new settings to the Cloud Clients, and some changes to the Objects parameters such as the demanded MIPS, RAM, Storage Size, Bandwidth and also changed the position to some far points that exceed the average values and checked if IoTSecurity was able to detect all the intrusions. After adding 50, 100 and 200 different new objects, the results were and are quite satisfying. The results are depicted in Figure 4.

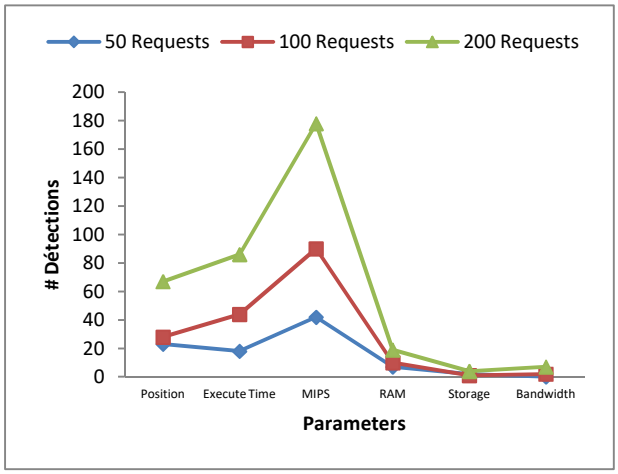

Fig. 4. Intrusion detection system based on spatial profile

We can notice that the detected objects are due to the abnormal changing occurred on the MIPS parameter because of the speed of execution that is changing radically in the ExecuteTime compared to the average values. The Storage Size, RAM and bandwidth parameters have only a small chance to occur as an Intrusion owing to the Cloud servers being as powerful and performant as they are supposed to be. Then we thought of leveling the stacks to another higher level, so we tested the intrusion detection on 500, 1000 and 5000 new objects. The time to detect intrusions on 5000 objects took a while (about 5 minutes of execution) due to the low performant computer, as a consequence, we conclude that our IoTSecurity is dedicated only for performant Cloud Servers with powerful CPU and much faster RAM. Concerning the intrusion Detection (figure 5), we can see in the that the MIPS is always giving much higher results of detected objects, and the Position, RAM, Storage Size and Bandwidth differential have only a small amount of detected objects as intruders.

978-1-5386-3269-7/17/$31.00 ©2017 IEEE

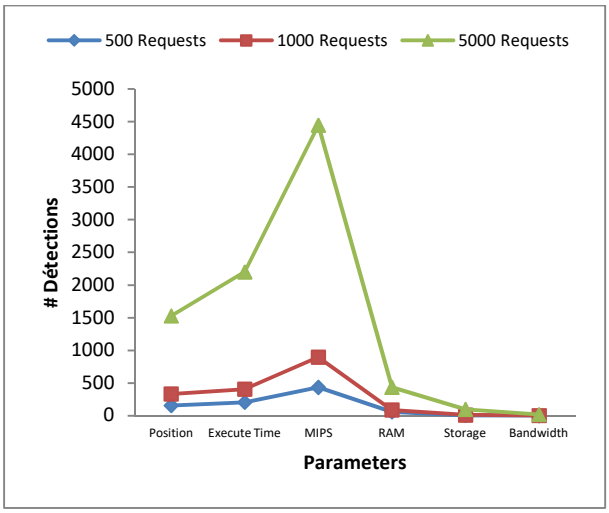

Fig. 5. Intrusion detection system for large requests

V. Conclusion and Perspectives

Due to the high evolvement of the number of devices connected to the Internet, and the high growing of the Internet Technology itself, we have to develop the whole Internet to another level. On the basis of this idea, the Internet of Things was created: where every object connects and without any human interference while keeping and saving the aspect of ubiquity. We have proposed, IoTSecurity, an intrusion detection system able to detect intrusions based on user profile. We exploited temporal and spatial profile. In the coming work, we intend to study the composition of a set of user profiles and the efficiency of this composition to better detect the intrusion in the CloudIoT environment.

References

[1] Rajkumar Buyya - Internet of Things (IoT): A vision, architectural elements, and future directions, 2013.

[2] R. Buyya, C. Yeo, S. Venugopal, J. Broberg, and I. Brandic - Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility, Future Generation Computer Systems, Elsevier, The Netherlands, June 2009.

[3] Thierry Evangelista - The IDS: The Intrusion Detection Systems in Computer Science, Dunod 2004.

[4] Rajkumar Buyya, Xiaofeng Wang - CloudEyes: Cloud-based malware detection with reversible sketch for resource-constrained internet of things (IoT) devices, University of Melbourne, 2016.

[5] Jon Oberheide, Evan Cooke, Farnam Jahanian - CloudAV: N-Version Antivirus in the Network Cloud, University of Michigan, July 2008.

[6] K. Ashton - That ‘‘Internet of Things’’ thing, RFiD Journal (2009).

[7] Dave Evans - L’Internet des objets Comment l’évolution actuelle d’Internet transforme-t-elle le monde ?, Cisco Internet Business Solutions Group, April 2011.

[8] H. Gupta, A. V. Dastjerdi, S. K. Ghoshy, and R. Buyya, “iFogSim: a toolkit for modeling and simulation of resource management techniques in internet of things,” Edge and Fog Computing Environments, pp. 1–22, 2016.

[9] D. Kelaidonis, P. Vlacheas, V. Stavroulaki, S. Georgoulas, K. Moessner, Y. Hashi, K. Hashimoto, Y. Miyake, K. Yamada and P. Demestichas, “Cloud Internet of Things framework for enabling services in Smart Cities”, in Designing, Developing, and Facilitating Smart Cities. Urban Design to IoT Solutions. Springer Smart City Technologies, 2017.

[10] E. Borgia, The Internet of Things vision: Key features, applications and open issues, Computer Communications 54 (2014) 1–31.

[11] Zarpelão B. B., Miani R. S., Kawakani C. T., Carlisto de Alvarenga S., A survey of intrusion detection in Internet of Things. Journal of Network and Computer Applications, 84, 2017, 25-37.

[12] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, M. Rajarajan, A survey of intrusion detection techniques in Cloud, Journal of Network and Computer Applications 36 (1) (2013) 42–57.

978-1-5386-3269-7/17/$31.00 ©2017 IEEE