Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

ARTICLE TYPE

Open Challenges in Vetting the Internet-of-Things

Zakaria Maamar[1] | Ejub Kajan[2] | Muhammad Asim[3] | Thar Baker[4]

[1]Zayed University, Dubai, UAE
[2]State University of Novi Pazar, Novi Pazar, Serbia
[3]National University of Computer and Emerging Sciences, Islamabad, Pakistan
[4]Liverpool John Moores University, Liverpool, UK

Correspondence
Zakaria Maamar
Email: zakaria.maamar@zu.ac.ae

Present Address
College of Technological Innovation, Zayed University, Po Box 19282, Dubai, UAE

Summary

Internet-of-Thing (IoT) is a rapid-emerging technology that exploits the concept of inter-network to connect things such as physical devices and objects together. A huge number of things (6.4 billion are in use in 2016) are already acting without direct human control raising a lot of concerns about the readiness and appropriateness of existing security practices, techniques, and tools to secure the data collected and protect people's private lives. As a first step, this paper presses the importance of having a dedicated process for vetting IoT (by analogy to vetting mobile apps) with focus on exposing things' vulnerabilities that could be the primary source of attacks. These vulnerabilities are identified according to things' duties decomposed into sensing, actuating, and communicating. A set of questions shed light on things' vulnerabilities per type of duty.

KEYWORDS:
Duty, Internet-of-Things, Mobile apps, Threat, Vetting, Vulnerability.

1 | MOTIVATIONS

In the 21st century, security, confidentiality, integrity, and privacy are prevalent concerns to the extent that the Information and Communication Technology (ICT) community is putting tremendous efforts into addressing them. The number of ICT-based misuse and fraudulent cases are on the rise calling for a serious revision of existing practices, techniques, and tools. One of these practices is to **vet** ICT applications prior to their integration into real (sometimes critical) business operations. Mobile apps exemplify ICT applications whose rapid and uncontrolled widespread has become a major concern to policy makers. As of March 2017, there were 2.8 million apps posted on *Google Play Store* and 2.2 million apps posted on *Apple's App Store*, the 2 leading app stores in the world (www.statista.com/topics/1002/mobile-app-usage).

In conjunction with the mobile apps "fever", we observe some early signs of another "fever" that the Internet-of-Things (IoT) could end-up catching. 6.4 billion connected things were in use in 2016, up 3% from 2015, and will reach 20.8 billion by 2020 according to Gartner (www.gartner.com/en/newsroom). The wireless world research forum also predicted that by 2017, there will be 7 trillion wireless devices serving 7 billion people leading to the formation of the IoT [1]. Are all these IoT-compliant things (things, for short) safe, secure, and trustworthy? Can we integrate them into critical systems? What are their vulnerabilities? To address these questions, **why don't we begin by vetting things like the ICT community does with mobile apps?** "*The IoT era not only brings new opportunities, but also presents an expanded attach surface, already being exploited by cyber criminals*" (go.armis.com/iot-security-buyers-guidev5) and "*IoT devices can and do get hacked regularly, and the consequences are severe*" (internetofthingsagenda.techtarget.com/blog/IoT-Agenda/How-can-anomalous-IoT-device-activity-be-detected).

The limited R&D initiatives on $\mathcal{V}$**etting IoT** ($\mathcal{V}$IoT) is accentuated by the heavy dependence of ICT practitioners on the security public claims of things' vendors. In a 2017 survey by Osterman Research, Inc. upon the request of Trustwave (www.trustwave.com/en-us/resources/library/documents/iot-cybersecurity-readiness), some key takeaways are that "*nearly three in five organizations can attribute some type of security incident to their IoT devices*" and "*IT departments charged with vetting IoT devices rely too heavily on IoT vendors' security claims and too little on internal testing, third-party testing and published reviews of the devices they connect to their networks*". Things in the IoT are not always collaborative, cooperative, and predictable [2]. By analogy to vetting mobile apps [3], we aim at developing the necessary concepts, principles, and techniques for vetting things by considering their intrinsic characteristics namely, reduced size, restricted connectivity, continuous mobility, limited energy, and

constrained storage. Our main objectives in the $\mathcal{V}$IoT project are (*i*) to define vetting in the context of IoT, (*ii*) to identify security vulnerabilities of things (focus of this paper), (*iii*) to define things' duties that could be subject to such vulnerabilities, and (*iv*) to develop and/or illustrate techniques that could address these vulnerabilities. The rest of this paper is organized as follows. Section 2 is a brief overview of vetting mobile apps and security of IoT. Section 3 discusses our vision of developing a comprehensive $\mathcal{V}$IoT process. Section 4 concludes the paper by presenting our current efforts into this process.

2 | BACKGROUND

In this section, we first define the concept of vetting mobile apps and then, discuss some initiatives related to security of IoT. We base our $\mathcal{V}$IoT process on vetting mobile apps' best practices and techniques.

2.1 | Vetting mobile apps in brief

According to the US National Institute of Standards and Technology (NIST) [4], there is a need to secure mobile apps from vulnerabilities and defects; apps are used by all people and all organizations. To satisfy this need, a strict vetting process would ensure that mobile apps comply with an organization's security requirements and are to a certain extent free of (serious) vulnerabilities. These requirements could be related to origin, data sensitivity, and target environment and could be split into (*i*) general with respect to some known standards and best practices like those specified by NIAP, OWASP, MITRE, and NIST, and (*ii*) specific with respect to some organizations' internal policies, regulations, and guidelines. Factors that cause app vulnerabilities include design flaws and programming errors, which could have been inserted intentionally or inadvertently [4]. Depending on the risk tolerance of an organization, some vulnerabilities might be more serious than others calling for a contextualized vetting process. Vetting could occur throughout an app's lifecycle that consists of development, acquisition, and deployment stages and would cover correctness testing, source and binary code testing, and static and dynamic testing.

Quirolgico et al. mention that millions of apps for mobile devices are out there through commercial stores and open repositories [3]. Because of their low cost and widespread, the threats of their vulnerabilities could be far greater than that of traditional computers. And, because some vulnerabilities of mobile apps are unique, Quirolgico et al. insist on the urgency of having a quick and cost efficient vetting process.

2.2 | Security of IoT in brief

Compared to vetting mobile apps (Section 2.1), there is a major gap in $\mathcal{V}$IoT. IoT mixes physical processes with digital (cyber) connectivity making it different from other software-related disciplines. The first set of references that we reviewed are more concerned with the security and privacy of IoT applications than by developing a comprehensive guide for vetting things that would reveal their vulnerabilities. And, the second set of references run tests to identify vulnerabilities of IoT devices that are already in operation. There is a consensus that IoT is vastly impacting the way we view, use, and interact with smart devices [5]. However, security remains a concern that could turn IoT misuse into a nightmare; such devices collect and use of a lot of personal data on their users. McKinsey argues that security may represent the greatest obstacle to IoT growth (www.mckinsey.com/industries/semiconductors/our-insights/security-in-the-internet-of-things). And, Creager discusses ways of detecting IoT devices' suspicious activities [6]. Devices are monitored for proper behavior, and those that show signs of having been interfered with can have their behaviors mitigated and security issue eliminated. Monitoring implies that devices are already in operation, which could be late since these devices were not vet.

In a 2018 report by KEYFACTOR (www.keyfactor.com), the authors discuss cases of IoT devices that have been subject to attacks although these devices were critical to humans' lives. Vetting IoT devices would have helped prevent or at least reduce such cases by exposing their vulnerabilities ahead of time. In the healthcare domain, the US Food and Drug Administration (FDA) recalled, in 2017, 465K pacemakers after discovering security flaws that could allow hackers to drain device batteries or send malicious instructions to modify a patient's heartbeat. Vetting pacemakers could have prevented such a serious case. Similar news is reported in the automotive industry when a Jeep Cherokee was hijacked turning off the transmission while the vehicle was on the freeway. Attacks also target robotic vehicles like drones and ground rovers [7]. Choi et al. consider robotic vehicles as a type of cyber-physical systems that consist of both cyber and physical components working jointly to support the vehicle's operations in the physical world. The authors enumerate many cases (e.g., GPS spoofing and ABS tampering) that show how vulnerable robotic vehicles are to attacks and suggest an invariant approach to address some vulnerabilities.

Out of the multiple references that we reviewed, the works of Palavicini Jr. et al. [8] and Siboni et al. [9] nicely overlap with our objectives. On the one hand, Palavicini Jr. et al. apply symbolic analysis to vet, in a semi-automated way, Industrial IoT (IIoT) firmware using *angr*, a UC Santa Barbara binary analysis framework [10], and *Mecanical Phish*, a component from the same university's cyber reasoning system, to perform semi-automated analysis

of IIoT. The authors mention that embedded systems and IIoT devices are rapidly increasing in number and complexity. As a result, cyber-physical attacks have become omnipresent causing economic and physical damages. They also mention that the firmware for these systems has become difficult to analyze when searching for malicious functionalities. Their approach consists of 3 steps: preparation of the firmware image for loading into the *angr* framework, emulation for verification of discovered vulnerabilities, and analysis of the firmware sample "angr style".

On the other hand, Siboni et al. discuss a security testbed for IoT devices that iTrust Lab has developed [9]. First they report on the experience of using an IoT search engine, SHODAN (www.shodan.io), to discover several vulnerabilities of IoT devices. It is worth noting that this experience targeted devices that were already in operation while we insist that vetting aims at detecting vulnerabilities prior to putting devices into operation (prevention is always better than correction). iTrust testbed emulates different types of testing environments that simulate the activity of multiple sensors and perform predefined and customized security tests along with advanced security testing analysis. Siboni et al. break down security aspects of IoT devices into 4 parts: device architecture that investigates attacks on hardware and software; network connectivity that investigates attacks on data distribution; data collection that investigates data collection with regard to privacy invasion and information theft; and, finally, countermeasures and mitigation that investigate how to reduce the security and privacy risks that IoT devices pose. The testbed's capabilities include initialization and detection, security tests, logging and analysis, usability, security aspects in terms of reliability, antiforensic, security, accountability, and finally adaptive in terms of scalability, performance, and flexibility.

3 | $\mathcal{V}$ETTING IOT-COMPLIANT THINGS

In this section we provide a definition of $\mathcal{V}$IoT, identify things' duties, and then discuss some vulnerabilities that could undermine the normal operation of these duties. A comprehensive $\mathcal{V}$IoT process is deemed necessary.

3.1 | Overview

According to Crews and Mangal [11], the massive arrival of smartphones and mobile apps has triggered a "mini-revolution" in the software engineering discipline. Some concepts, principles, and practices like those related to testing have been reviewed, for example. Touchscreen gestures, location awareness, and orientation need to be tested differently. The same is valid when testing IoT smart devices and, also, vetting them. We expect that IoT features such as reduced size, restricted connectivity, continuous mobility, limited energy, constrained storage, and additional features that Kamrani et al. [12] discuss, will trigger a similar "revolution".

By analogy to the NIST definition of app vetting process [4], $\mathcal{V}$IoT process would be a sequence of stages that an organization would perform to declare if a thing is "clean" (in terms of safety, security, trustworthiness, etc.) with respect to some IoT safety and security requirements/guidelines/policies. Our sequence of 4 stages represented in Fig. 1 would consist of defining things' duties that would be subject to vetting, identifying the vulnerabilities that would affect these duties, analyzing the impact of these vulnerabilities on these duties, and developing guidelines and/or recommending techniques to address these vulnerabilities. In this paper, we discuss stage 2.

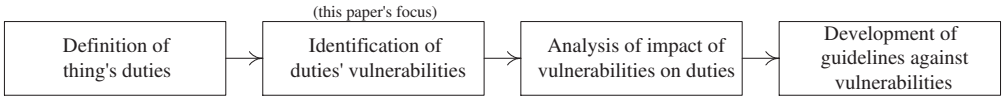

FIGURE 1 4 stages of the proposed $\mathcal{V}$IoT process

3.2 | Duties of things

In a previous work [13], we identified 3 single duties that would capture a thing's capabilities in terms of *sensing* (collecting/capturing data), *actuating* (processing/acting upon data), and *communicating* (sharing/distributing data). A duty is either enabled or disabled ((0,1) in Fig. 2) according to the requirements and needs of the under-development IoT applications.

In terms of duties, a thing senses the cyber-physical surrounding so, that, it generates (raw) data; a thing actuates data including those that are sensed; and a thing communicates with the cyber-physical surrounding the sensed and/or actuated data. Accepting data and/or commands from external parties (e.g., other things) is also taken care by the communicating duty but is not further discussed in this report. It is worth noting that

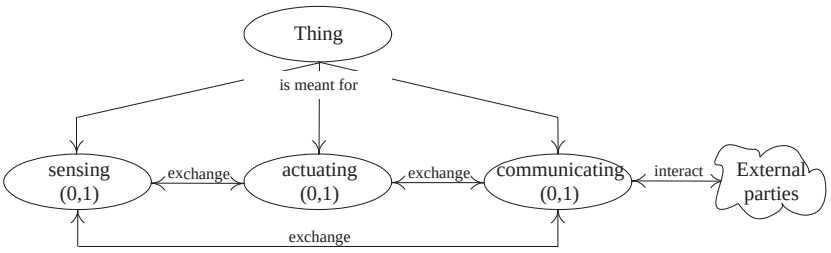

FIGURE 2 Duties associated with a thing

a thing's sensing, actuating, and communicating duties can be composed together as per the following 4 representative cases (other cases like Communicating→Actuating and Communicating→Actuating→Sensing are not discussed):

1. Sensing→Actuating→Communicating: sensed data are passed on to actuating; and the data that result from actuation are passed on to communicating for distribution.
2. Sensing→Actuating: sensed data are passed on to actuating; and the data that result from actuation are finals.
3. Sensing→Communicating: sensed data are passed on to communicating for distribution.
4. Actuating→Communicating: data that result from actuating are passed on to communicating for distribution.

3.3 | Vulnerabilities of things

As per Fig. 1, our $\mathcal{V}$IoT process proceeds with the definition of things' duties and then, identification of potential vulnerabilities that could impact the completion of these duties. Although some thing vulnerabilities that could constitute sources of attacks are already identified in the context of the OWASP IoT project (www.owasp.org), we find these vulnerabilities generic and not focused on IoT features nor things' duties. To address this limited focus, we present in the following some questions that would help identify vulnerabilities of some duties, whether simple or composite. By addressing these questions, we aim in the future at recommending techniques and/or practices that would mitigate such vulnerabilities (stage 4). Due to lack of space, only sensing and actuating as single duties, and only sensing → actuating as a composite duty, are discussed.

- Questions related to vetting sensing include, but are not limited to:
    1. Does sensing target living (e.g., persons) and/or non-living things (e.g., rooms)? And, what are we sensing? Ambient temperature, wind speed, heartbeat, etc.? *Vulnerabilities*: sensing things without their approval and sensing something different than what is claimed.
    2. Does sensing target indoor, outdoor, or both? *Vulnerabilities*: sensing something different than what is claimed.
    3. What is the frequency of sensing such as continuously, at regular intervals, or trigger-based? *Vulnerabilities*: changing sensing frequency without approval and sensing differently from what is claimed.
    4. Who does configure sensing in terms of frequencies, service periods, authorized recipients, etc.? And, does configuration have to happen from a specific location and/or using a specific device? *Vulnerabilities*: unauthorized person proceeds with configuring sensing using a non-acceptable device and/or from an unacceptable location.
    5. What is the resource consumption level of sensing? And, is there any threshold that would indicate over-consumption and hence, trigger alarms? *Vulnerabilities*: tampering sensing's approved resource consumption-level so, that, over consumption goes unnoticed.
    6. Are there traces of tracking sensing using logs, for example? If yes, how are these traces safeguarded? *Vulnerabilities*: altering tracking traces of sensing and distributing traces to unauthorized parties.
- Questions related to vetting actuating include, but are not limited to:
    1. Can a thing cancel and/or compensate the outcomes of actuating? If yes, does it need any approval? *Vulnerabilities*: canceling and/or compensating actuating outcomes without approval.

2. What is the frequency of actuating such as continuously, at regular intervals, or trigger-based? *Vulnerabilities*: changing actuating frequency without approval and actuating differently from what is claimed.
3. Who does configure actuating in terms of frequencies, service periods, etc.? And, does configuration have to happen from a specific location and/or using a specific device? *Vulnerabilities*: unauthorized person proceeds with actuating configuration using a non-acceptable device and/or from an unacceptable location.
4. What inputs does actuating require? And, what outputs does actuating produce? *Vulnerabilities*: altering inputs and outputs purposely.
5. Are there traces of tracking actuating using logs, for example? If yes, how are these traces safeguarded? *Vulnerabilities*: altering tracking traces of actuating and distributing traces to unauthorized parties.
6. What is the resource consumption level of actuating? And, is there any threshold that would indicate over-consumption and hence, trigger alarms? *Vulnerabilities*: tampering actuating's approved resource consumption-level so, that, over consumption gets unnoticed.

- Questions related to vetting sensing $\rightarrow$ actuating include, but are not limited to:

1. Are sensing and actuating frequencies synchronized, or not? *Vulnerabilities*: changing frequencies of sensing and actuating without approval or making sensing and actuating (un)synchronize though it is not mandatory.
2. How and by whom sensing and actuating are configured? And, does configuration have to happen from a specific location using a specific device? *Vulnerabilities*: unauthorized person proceeds with sensing and actuating configuration using a non-acceptable device and/or from an unacceptable location.
3. Does actuating consider all sensed data as is or does actuating pre-process sensed data before? *Vulnerabilities*: altering sensed-data outputs and therefore sensed-data inputs to actuating.
4. What is the resource consumption level of sensing and actuating? And, is there any threshold that would indicate over-consumption and hence, trigger alarms? *Vulnerabilities*: tampering sensing and actuating's approved resource consumption-level so, that, over consumption gets unnoticed.
5. Are there traces of tracking sensing and actuating using logs, for example? If yes, how are these traces safeguarded? *Vulnerabilities*: altering tracking traces of sensing and actuating and distributing traces to unauthorized parties.

4 | CONCLUSION

IoT systems encompass many heterogeneous things that are connected together via Internet technologies and protocols to communicate and exchange data with each other and their surrounding environment. These complex systems have some associated vulnerabilities and issues that, once occurred, are destructive. By analogy to vetting mobile apps, it is extremely important to address the lack of principles, techniques, and tools for vetting things in preparation for their integration into mission-critical systems, for example. Things have got vulnerabilities that should be "discovered" through proper vetting. Unfortunately, this is not happening!! Rather than sensing a turbine's steam level, a thing could collect some sensitive data about the turbine without users' knowledge and leak these data to third parties. In this paper, we illustrated the first steps of our $\mathcal{V}$IoT process that should help promote a better and safe IoT. We defined things' duties that are subject to vetting and identified their vulnerabilities. In compliance with Fig. 1 our ongoing work is to analyze the impact of these vulnerabilities on things' duties and to develop guidelines against these vulnerabilities prior to thing deployment into real-world scenarios.

Author contributions

All authors contributed to the text.

Financial disclosure

N/A.

Conflict of interest

The authors declare no potential conflict of interests.

References

1. Razzaque M, Milojevic-Jevric M, Palade A, Clarke S. Middleware for Internet of Things: A Survey. *IEEE Internet of Things Journal* 2016; 3(1).
2. Chen C, Helal S. A Device-Centric Approach to a Safer Internet of Things. In: Proceedings of NoME-IoT'2011 Workshop. ; 2011; Beijing, China.
3. Quirolgico S, Voas J, Kuhn R. Vetting Mobile Apps. *IT Professional* 2011; 13(4).
4. Ogata M, Franklin J, Voas J, Sritapan V, Quirolgico S. Vetting the Security of Mobile Applications. DRAFT NIST Special Publication 800-163 (Revision 1); Downloaded in September 2018.
5. Council YE. 10 Big Security Concerns about IoT For Business (And how to Protect Yourself). July 2018 (checked out in October 2018). www.forbes.com/sites/theyec/2018/07/31/10-big-security-concerns-about-iot-for-business-and-how-to-protect-yourself/#1ebe58897416.
6. Creager L. How can Anomalous IoT Device Activity be Detected?. tinyurl.com/y9v5lgfq; Downloaded in November 2018.
7. Choi H, Lee W, Aafer Y, et al. Detecting Attacks Against Robotic Vehicles: A Control Invariant Approach. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS'2018). ; Toronto, ON, Canada.
8. Palavicini Jr. G, Bryan J, Sheets E, Kline M, San Miguel J. Towards Firmware Analysis of Industrial Internet of Things (IIoT) - Applying Symbolic Analysis to IIoT Firmware Vetting. In: Proceedings of the 2nd International Conference on Internet of Things, Big Data and Security (IoTBDS'2017). ; 2017; Porto, Portugal.
9. Siboni S, Sachidananda V, Meidan Y, et al. Security Testbed for Internet-of-Things Devices. *IEEE Transactions on Reliability* 2018.
10. Shoshitaishvili Y, Wang R, Salls C, et al. SoK: (State of) The Art of War: Offensive Techniques in Binary Analysis. In: Proceedinmgs of the Symposium on Security and Privacy (SP'2016). ; 2016; San Jose, CA, USA.
11. Crews B, Mangal S. IoT and it's Impact on Testing. www.getzephyr.com/resources/whitepapers/iot-and-its-impact-testing; Downloaded in November 2018.
12. Kamrani F, Wedling M, Rodhe I. Internet of Things: Security and Privacy Issues. tech. rep., FOI Swedish Defence Research Agency, Defence and Security, Systems and Technology; 2019.
13. Qamar A, Muhammad A, Maamar Z, Baker T, Saeed S. A Quality-of-Things Model for Assessing the Internet-of-Thing's Non-Functional Properties. *Transactions on Emerging Telecommunications Technologies* 2019 (forthcoming).