

Towards IoT Cybersecurity Modeling: From Malware Analysis Data to IoT System Representation

A. Rodríguez-Mota*, P.J. Escamilla-Ambrosio[†], J. Happa[‡], J.R.C. Nurse[‡]

*Instituto Politécnico Nacional, ESIME, Zacatenco, Mexico City, Mexico 07738

Email: armesimez@gmail.com

[†] Instituto Politécnico Nacional, Centro de Investigación en Computación, Mexico City, Mexico

Email: pescamilla@cic.ipn.mx

[‡]Department of Computer Science,
University of Oxford, UK

Email:jassim.happa@cs.ox.ac.uk, jason.nurse@cs.ox.ac.uk

Abstract—The heterogeneous nature of the Internet of Things (IoT) represents a big challenge in many different technical and scientific areas, among them Security. In this sense, security becomes an extremely complex problem as it is present in every aspect of the IoT ecosystem, from sensors and data acquisition hardware to front-end software applications and sophisticated user devices. This complexity expands as there is not consensus among all stakeholders towards the definition of general technical standards, specifications, system representations and use policies. In this context, this paper presents a state of intention for a research project oriented to construct a set of tools to characterize security attack surfaces for IoT systems solutions. The proposed research includes the development of a visual grammar aimed to depict IoT systems at a high-abstraction level together with the construction of objects profiles, which in conjunction will provide building blocks and mechanisms to evaluate or identify insecure IoT scenarios.

I. INTRODUCTION

IoT not only has the same security issues as sensor networks, mobile communications networks and the Internet, but also has its specialties such as privacy issues, different authentication and access control network configuration issues, information storage and management and so on [1]. The interconnected nature of IoT devices means that every poorly secured device that is connected online potentially affects the security and resilience of the Internet globally. This challenge is amplified by other considerations like the mass-scale deployment of heterogenous IoT devices, the ability of some devices to automatically connect to other devices, and the likelihood of fielding these devices in insecure environments [2]. When thinking about Internet of Things devices, it is important to realize that, like any other devices, security cannot be guaranteed. IoT security is not a binary proposition of secure or insecure. Instead, it is useful to conceptualize IoT security as a spectrum of device vulnerability. The spectrum ranges from completely unprotected devices with no security features at all to highly secure systems with multiple layers

of security features [2]. The overall security and resilience of the IoT is a function of how security risks are assessed and managed. Security of a device is a function of the risk that a device will be compromised, the damage such compromise will cause, and the time and resources required to achieve a certain level of protection [2].

Several factors influence the risk assessment and mitigation calculation. Factors include having a clear understanding of the present security risk and the potential future risks; the estimated economic and other costs of harm if the risks are realized. While this kind of security tradeoff are often made from an individual user or organizational perspective, it is also important to consider the interrelatedness of IoT devices as part of a larger IoT ecosystem. The networked connectivity of IoT devices means that security decisions made locally about an IoT device can have global impacts on other devices [2].

There are several ways in which we can consider the security needs within the IoT space. One of the most well regarded IoT security architectures has been proposed by Jing et al [1]. In their architecture, they aim to represent key aspects of interest, and divide the IoT structure into three layers: the perception layer, the transportation layer, and the application layer. These layers are then broken down further where: the perception layer is composed of perception nodes and perception network; the transportation is split into access network, core network, and LAN; and the application layer is divided into application support layer and IoT applications [1]. Each of these layers allows the security features and requirements to be characterised in more detail and as necessary depending on the target level of the device.

The remainder of this article is as follows. In Section II we discuss the topic of visualising risks and security, which is a core topic of our proposed work. Section III begins presenting a system proposal for IoT modeling, providing a general description of its main components, focusing on the discussion of an specific process which reuses malware analysis data

to produce a visual representation of the physical objects in the system. In this case, object characterization is focused on Android OS powered devices as Android has become a major player in the mobile devices market and we already have developed a Web system capable of extracting static features from application files. Finally, Section IV presents our conclusions and highlight some future research areas.

II. VISUALISING RISK

There has been a noteworthy amount of research conducted on the topic of visualising risks, particularly in the security context. One of the most novel of these works is that by Hogganvik and Stlen [3]. They propose and evaluate a graphical approach to identify, explain and document various security threats and risk scenarios. At the core of the approach is a rigorously defined conceptual model that characterises crucial aspects of security and risk. Another relevant article to this topic of visualising risks is that of Li et al. [4]. They aim to add value by outlining several methods for visualising information security threats; the goal is to create a flexible basis for creating semantically rich threat visualisation diagrams that can be used for various situations. While in general either of these two approaches might be applied to IoT security, there are numerous challenges in modelling such a large and diverse ecosystem of devices.

Many IoT devices, such as sensors and consumer items, are designed to be deployed at a massive scale that is orders of magnitude beyond that of a traditional Internet-connected devices. As a result, the potential quantity of interconnected links between these devices is unprecedented. This alone poses security threats we have not seen before, simply because society has never had such a massive influx of connected devices before. Therefore, existing tools, methods, and strategies associated with IoT security may need new consideration [2].

In this sense this paper presents a system capable to produce IoT systems visual representations. The system will be based on a new visual grammar to help identifying new attack surfaces or potential attack vectors in an existing infrastructure. This represents a new methodology to do attack surfaces and vulnerability detection, with the IoT infrastructure in mind. Additionally, this work also aims to explore the possible potential impact of combining conceptual representations of IoT threats (modeling) and empirically-driven IoT systems characterizations towards designing better security assessment strategies and methodologies. A general conceptual representation of the proposed development is depicted on Figure 1. In a general way, this three-tiered structure presents the relationships among the models of an IoT system (high level representation) and the details of its implementation (low level representation). It is considered that such structure provides a flexible scheme which can be followed either in a top-bottom or bottom-up fashion. In other words, the three tier structure would allow users to construct physical IoT systems instances from high level models (top-bottom approach) or from existing physical IoT systems produce accurate models (bottom-up approach).

Visual grammars can provide information about objects (the "what"), the structures they form (the "how are the objects

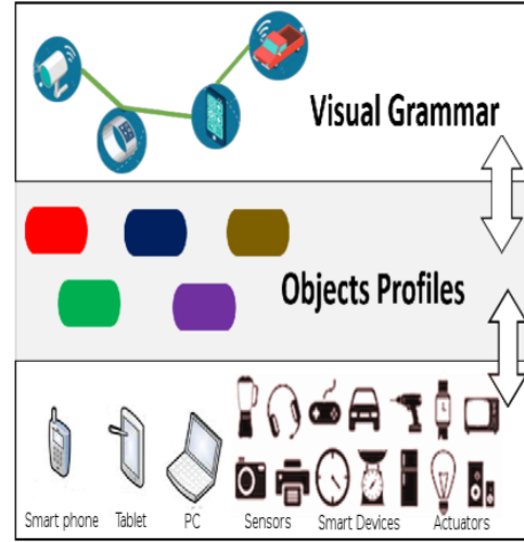


Fig. 1. General System Conceptual representation.

connected?"), their activities ("the processes they do with each other in this structure") and their relationships ("the way the objects, activities and structures relate to each other") [5]. Using our three-tiered conceptual representation, see Figure 1, we go from an empirically-driven representation of the IoT environment (with logs from the devices themselves) to a capability-driven representation (with abstract views of what the devices are capable of doing). In other words, we go from a representation driven by data logs, and to a representation that describes what the IoT devices can do partly derived from the data, but also partly derived from the device specifications (if these are available)

Our visual grammar is structured in a similar manner to any language: we define a basic formal syntax that is able to convey:

- communication patterns – "what protocols are used between my devices, and how"),
- infrastructure topology – "what is the abstract shape of how are my IoT devices link to each other, and how to they link to the internet?",
- cyber-physical capabilities – "what does the device do in the real world? why does it exist? "

Our approach borrows from graph theory: we use an undirected graph to represent our topology, with each vertex encoding properties about the IoT device (the aforementioned cyber-physical capabilities) and the edges encoding information about cyber-physical capabilities as well as communication patterns. In essence, we propose what visual grammars already deliver, but with syntax that is useful to obtain situational awareness. If we can abstractly represent all paths IoT devices can take, map their properties, and represent this information in a useful manner to users, they should be able to use this insight to better secure their IoT network.

One of the major challenges of the development of the proposed visual grammar relates to the capacity to represent objects and their relationships in a high accurate form. These

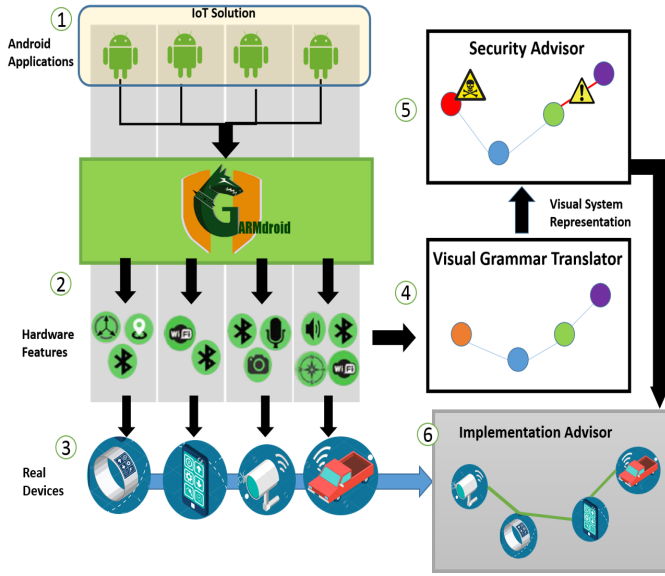


Fig. 2. Conceptual representation of the system general architecture.

detailed representations are key aspects to model real attack surfaces on the systems. In this sense, the rest of this paper briefly describes the modules of the proposed system with special emphasis on describing the empirically-driven object characterization process that will help to characterize real objects. Furthermore, considering the case where smart mobile devices powered by the Android OS are included as elements of IoT systems, details are provided about the object characterization process, supported by previous results obtained by using a software tool, named GARMdroid, developed as part of an ongoing research project in the Android malware analysis and detection area. Therefore, discussion continues by briefly introducing GARMdroid [6] and describing how the malware analysis data it extracts and process can be used to provide an object characterization.

III. USING ANDROID MALWARE ANALYSIS DATA FOR PHYSICAL IoT OBJECTS CHARACTERIZATION

In the context of the project described in this paper, GARMdroid provides a mechanism to identify general characteristics of Android powered devices. On a first approach, it has been set an operational scheme where objects participating on an IoT system implementation can be characterized by reviewing static analysis data extracted using GARMdroid from a set of Android applications. This scheme is broadly illustrated on Figure 2.

Observing the proposed scheme, see Figure 2, 6 major processes can be identified, numbered from 1 to 6 in the figure. At first, the analysis starts from an IoT Solution system implemented as a set of different Android applications. These applications are entered to GARMdroid which infers the hardware features requested by the application from the static information contained in the application's AndroidManifest file. This features, together with information about the Android OS and Java version numbers, are used to create objects profiles that match the applications requirements, this step is

labeled as number two in the figure. The objects profiles are used to identify real devices already known by the system (stored on a data base), this is step 3. During step number 4, the objects profile identified are handed in to the Visual Grammar Translator which task is to construct a model from the given objects models. Once the model is obtained there is a further step, labeled as number 5, where the model is analyzed from the security point of view base on the hardware and services requirements previously identified. Finally, the output from the Security Advisor block is joined with the real devices profiles identified during step 3 and are feed into the Implementation Advisor which aims to provide a system specification considering the best security practices.

1) *Android malware data analysis and the empirically-driven object characterization process:* During the recent years Android has become one of the major stakeholders in the Mobile Operating Systems (OS) market. Its openness, together with its free distribution scheme, has played an important part in the fast adoption of this OS. Additionally, with the advent of cheaper, smaller, and more reliable sensors and devices, Android impact has reached other areas such as the wearable devices space, making Android a very attractive target for cyber-attacks. In this sense, it is important to recall that the presence of mobile devices as elements of IoT systems is every time more common. Moreover, the limitation on power, computing capacity and other factors, will require significant changes in the way that these devices are protected, as traditional concepts of firewalls and anti-malware are unlikely to translate well to their capabilities. In other words, every connected device is a potential security weakness that could attack or co-opt other devices connected to it [7].

GARMdroid, is a 2-hybrid Android malware analysis and detection system under development. The term 2-hybrid implies both a local(host) - remote(server/cloud) implementation and a static-dynamic analysis approach [8]. Its name is the result of the fusion of the words Garm and Android, where Garm is described in the Norse mythology as a watchdog that guards Hells gate [9]. Broadly, the proposed system is categorized as 2-hybrid to reflect the fact that it integrates static and dynamic malware analysis techniques, and local and remote analysis execution [10].

GARMdroid is based on the capabilities provided by the Android SDK tool set, specifically the Android Asset Packaging Tool (AAPT) which is contained as part of the *platform tools* set. In this implementation clients can upload malware samples and request analysis via a Web interface [6]. Figure 3 shows the main page of the system from where users can upload files and see the results after file processing. Once the application file is processed the tool displays the name, MIME type, size and md5 hash value of the file. Additionally, permissions and features are identified and displayed. In the case of permissions, Figure 4a, it has been selected to visualise the requested permissions as a matrix of dots where permissions requested by the application under analysis are indicated as red dots. Features have been represented as icons in order to facilitate visualisation: Audio, Bluetooth, Camera, Infrared, Location, Microphone, NFC, Sensors (Accelerometer, Barometer, Compass, Gyroscope, Light, Proximity, Step Counter,

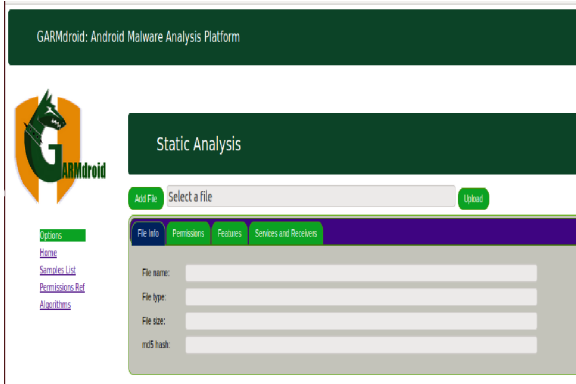


Fig. 3. Garmdroid's Welcome Page.

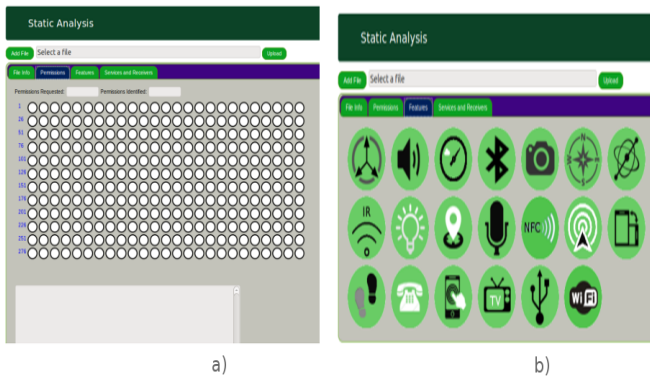


Fig. 4. a) Permissions tab, additionally to the dot matrix representation, in which hovering over a circle provides the full permission name; a textbox element at the bottom of the tab also displays the identified permissions; b) Features tab, representing hardware features as icons which change its background color to red if they are requested by the file under analysis.

Step Detector), Screen, Telephony, Television, Touchscreen, USB and WiFi, see Figure 4b.

It is important to notice the key role of permission declaration in Android applications for defining hardware features and security risks. In this sense, it is useful to observe Table I, which presents some permissions that can harm the functionality of other applications, operating systems or hardware sensors, in order to gain an insight into the kind of information they can provide about devices and security aspects.

Figure 5 presents an example of the output provided by GARMDROID when a Hardware-Tester application has been provided. In this case, hardware features identified are highlighted by a red background. It is important to notice that the described process provides data that represents a form to characterize physical objects belonging to an IoT solution having only access to the Android applications apk files. In other words, it is possible to obtain a model of a system, from where to define or test security schemes and policies, by having only access to the apk files of the software of the system.

2) *Characterization of real devices and the Implementation Advisor:* Once a device has been characterized, based on the features extracted from the application software, these hardware requirements are compared against a set of devices

TABLE I
A LIST OF SOME ANDROID PERMISSIONS THAT CAN HARM THE FUNCTIONALITY OF OTHER APPLICATIONS, OPERATING SYSTEMS OR HARDWARE SENSORS, ADAPTED FROM [11].

Permission	Related Risks
CHANGE_NETWORK_STATE	Can change whether or not the device is connected to a network.
KEYGUARD	Can keep the Android device unlocked and unprotected, causing unwanted calls.
MODIFY_AUDIO_SETTINGS	May affect the usability of the device or impact other applications.
SET_TIME_ZONE	May affect the usability of the device or impact other applications.
WRITE_EXTERNAL_STORAGE	May result in a number of affects being realized including harming the actual memory of the device. Many writes and deletes may break memory segments. An application may fill the devices memory storage such that a victim would be unable to add more data or install required applications.
WRITE_CONTACTS	Similar effects as WRITE_EXTERNAL_STORAGE. By adding contacts malware could trick the user into calling unwanted numbers, change phone numbers of certain contacts or by adding contacts malware could fill the space for contacts on a SIM card.
KILL_BACKGROUND_PROCESSES	It is not necessarily malicious and would not do much harm, but it could impact the usability of the device, for example by killing processes without user's consent.

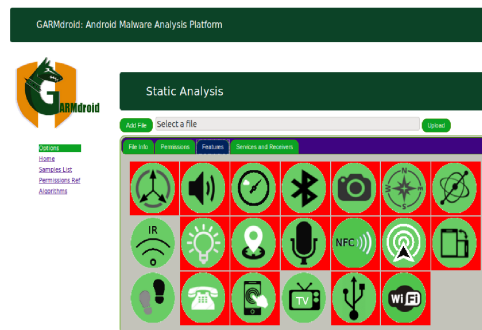


Fig. 5. Features requested by a sample Hardware-Tester application.

profiles in order to identify one or more real devices which comply with the identified requirements. These profiles are stored by the Implementation Advisor. Each profile is created by analyzing the manufacturers specifications or by direct

hardware evaluation.

3) *The Visual Grammar Translator*: After an object has been characterized and mapped to one or more real devices, information is used to provide a higher level representation. Although there is not a standard modeling language specific for IoT systems some approaches exist. Therefore, as a proof of concept, in this project it has been decided to process the hardware features information obtained and produce an initial representation of the related devices. As a first approach, the System Modeling Language's (SysML) Block Definition Diagram (BDD) notation was used [12]. SysML's BDD notation was selected taking into account the growing adoption of SysML for modeling mechatronic systems and its relationship with other initiatives, such as the Model Integrated Mechatronic (MIM) [13], 3+1 SysML View-Model [14] and UML4IoT [15] approaches.

The Block Definition Diagram is the most common SysML type of diagram. When using BDD notation is possible to display various kinds of model elements and relationships that express information about the structure of a system. Blocks can represent any level of the system hierarchy including the top-level system, a subsystem, or logical or physical component of a system or environment, as well as software entities. Blocks are modular units of system descriptions, each of them can define a collection of features to describe a system or other element of interest. These may include both structural and behavioral features, such as properties and operations, to represent the state of the system and behavior that the system may exhibit [13].

Additionally, SysML blocks provide a general-purpose capability to model systems as tree of modular components. The notation is based on UML classes notation extended by UML composite structures. However, some capabilities available for UML classes, such as more specialized forms of associations, have been excluded from SysML block to simplify the language. SysML Blocks notation also extends the capabilities of UML classes and connectors with reusable forms of constraints, multi-level nesting of connector ends, participant properties for composite association classes, and connector properties [12].

In the first case, based on the data presented previously in Figure 5, a SysML BDD diagram for the Hardware-Test sample was obtained, see Figure 6.

In a second case, Figure 7 and 8 present the requested hardware features extracted by GARMDROID from the apk file, and the BDD SysML representation obtained based on this information, respectively.

4) *The Security Advisor*: This module stores a set of security guidelines related to individual elements as well as groups of them. These guidelines are extracted from general security guidelines documents, manufactures manuals and implementers reports. Thus, once the system has identified and characterized the set of possible devices that form the IoT system under analysis, the Security Advisor (SA) identify from its set of security guidelines the potential security vulnerabilities and produce a report. The security risk report is used to produce a risk visual representation model of the system. Later on, the SA searches for a set of security policies that

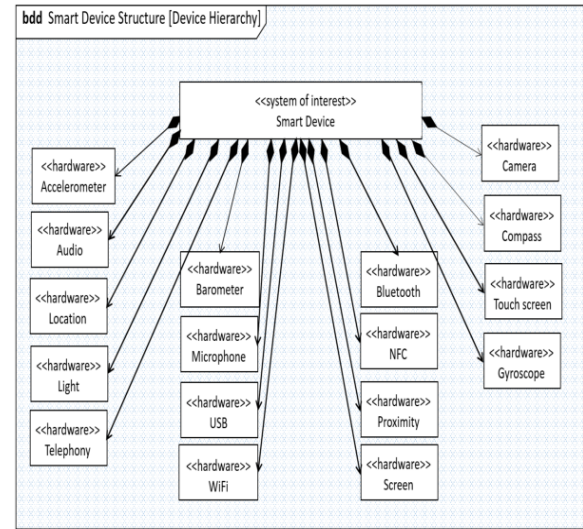


Fig. 6. SysML Block Definition Diagram for the Hardware-Test app.

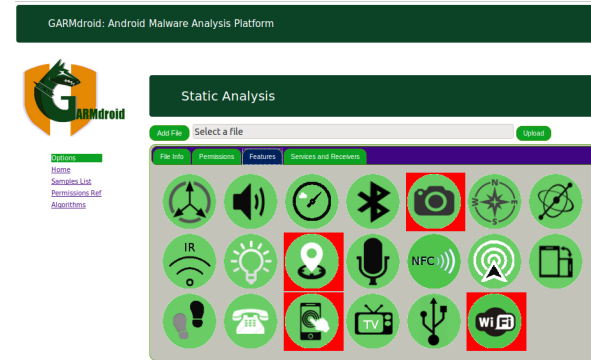


Fig. 7. Features requested by a lighting app.

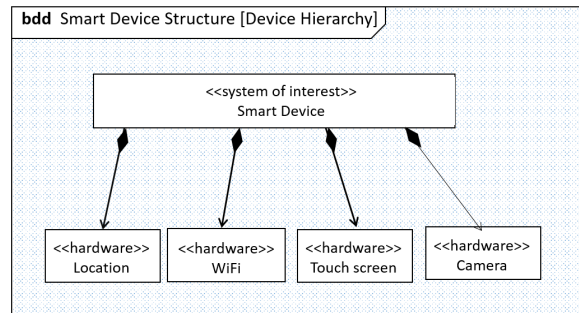


Fig. 8. SysML Block Definition Diagram for the lighting sample app.

better conform to the modeled system in order to provide one or various security profiles, which are available to the users as textual reports and visual models.

IV. CONCLUSION

The heterogeneous nature of the IoT represents a big challenge for the different stakeholders involved in the development of the supporting technology and regulations. In this sense, there is a big research opportunity in different areas, among them Security. Consequently, in this paper it has been presented a novel approach to security analysis for the IoT systems, involving the development of a visual grammar for the representation of IoT elements and their integration, a mechanism to identify real scenarios where models can be tested, and a strategy to provide security solutions.

We understand the enormous set of challenges involved in the development of the proposed project and it is also clear that further development is required. However, it is also clear that our current developments may provide a good insight for other researchers and people involved in the development of the IoT paradigm. In this sense, it is expected that the further development of GARMDROID, together with other tools we are currently developing, would provide useful data for developing IoT understanding.

It has to be observed that data is a major asset of the proposed system as modeling is based on manufacturers specifications, security guidelines and policies, and software and hardware analysis reports. In this sense, in a first approach to model validation it has been proposed the construction of a set of test systems from where models will be obtained and security tested. Additionally, a set of security test will be provided to validate the security policies suggested by the system.

Additionally, it is expected that the development of the proposed visual grammar will attract attention from industry, technologists, researches and systems integrators in such a way that many ideas can converge at this project helping us to move forward in the characterization and understanding of security for the Internet of Things.

ACKNOWLEDGMENT

This work was supported by the Mexican National Council of Science and Technology (CONACYT) under Grant 216747 and in part by IPN under Project SIP-20161697.

REFERENCES

- [1] Q. Jing and A. V. Vasilakos and J. Wan Lu and D. Qiu, *Security of the Internet of Things: perspectives and challenges* *Wireless Networks*, 20, 2481-2501, 2014.
- [2] K. Rose and S. Eldridge and L. Chapin, *The Internet of Things: An overview*, The Internet Society (ISOC), 2015.
- [3] I. Hogganvik and K. Stølen, *A graphical approach to risk identification, motivated by empirical investigations*. In International Conference on Model Driven Engineering Languages and Systems (pp. 574-588). Springer Berlin Heidelberg, October, 2006.
- [4] E. Li and J. Barendse and F. Brodbeck and A. Tanner, *From A to Z: Developing a Visual Vocabulary for Information Security Threat Visualization*. In International Workshop on Graphical Models for Security (pp. 102-118). Springer International Publishing, June, 2016.
- [5] C. Leborg. *Visual grammar*, Princeton Architectural Press, 2006.

- [6] A. Rodríguez-Mota and P.J. Escamilla-Ambrosio and J. Happa and E. Aguirre-Anaya, *GARMDROID: IoT Potential Security Threats Analysis through the Inference of Android Applications Hardware Features Requirements*, AFI 360 Conference Track on Future Internet and Internet of Things Applications, 2016.
- [7] the Policy and Research Group of the Office of the Privacy Commissioner of Canada *The Internet of Things. An introduction to privacy issues with a focus on the retail and home environments*. February, 2016. https://www.priv.gc.ca/information/research-recherche/2016/iot_201602_e.pdf
- [8] A. Rodríguez-Mota and P.J. Escamilla-Ambrosio and S. Morales-Ortega and M. Salinas-Rosales and E. Aguirre-Anaya, *Towards a 2-hybrid Android Malware Detection Test Framework*, 2016 International Conference on Electronics, Communications and Computers (CONIELECOMP), Cholula, 2016, pp. 54-61.
- [9] Wikipedia, *Garmr*. Wikipedia, <https://en.wikipedia.org/wiki/Garmr>, 15 12 2015.
- [10] A. Rodríguez-Mota; P. J. Escamilla-Ambrosio; E. Aguirre-Anaya; R. Acosta-Bermejo; L. A. Villa-Vargas, *Improving Android Mobile Application Development by Dissecting Malware Analysis Data*. 2016 4th International Conference in Software Engineering Research and Innovation (CONISOFT), 2016, pp. 81 - 86.
- [11] Milosevic, N. *Android Security: Malicious use of Android permissions*, Digital Forensics Magazine. Issue 18, February, 2014, pp. 28-31.
- [12] OMG *OMG System Modeling Language (OMG SysML)*. Version 1.4. September 2015. <http://www.omg.org/spec/SysML/1.4/PDF/>
- [13] Thramboulidis, K. *Model Integrated Mechatronics: An Architecture for the Model Driven Development of Mechatronic Systems*, 2nd IEEE International Conference on Mechatronics, pp. 497-502, Istanbul, Turkey 2004.
- [14] Thramboulidis, K. :The 3+1 SysML View-Model in Model Integrated Mechatronics. In: Journal of Software Engineering and Applications, Vol. 3 No. 2, 2010, pp. 109-118.
- [15] Thramboulidis, K., Christoulakis, F. : UML4IoTA UML-based approach to exploit IoT in cyber-physical manufacturing systems. In: Computers in Industry. <http://www.sciencedirect.com/science/article/pii/S016636151630094X>, 10 05 2016.