Enviado: 06/10/2022 *Recibido: 09/10/2022* *Aceptado: 20/11/2022*

UNIVERSIDAD DISTRITAL FRANCISCO JOSE DE CALDAS

VISIÓN ELECTRÓNICA

https://doi.org/10.14483/issn.2248-4728

VISIÓN ELECTRÓNICA

A RESEARCH VISION

Analysis of the unauthorized extraction of personal data from an IoT device

Análisis de la extracción de datos personales sin autorización de un dispositivo IoT

Juan Carlos Díaz[1] , Dorotea Zunino[2] , Giovanna Nicolino[3]

Abstract

In this paper, we work on the possibility that an IoT device sends user data without explicit permission. A prototype was built that receives the user's first and last name through a BlueTooth connection, and to that is attached the GPS location of the IoT device in the place it is located and transmits it to the server, without being informed at any time to ask the user for permission. The test was done with a tablet without a geolocator (since geolocation was provided by the IoT prototype with a GPS module) and as a result, several transmissions were made to a remote server with the position and name of the user, proving that it was possible to achieve these data, (which traditionally requires permissions to access) and send it without user permission.

Keywords: Authorization, Geolocation, GPS, IoT, Privacy, WPS.

[1] Engineer, Universidad de Palermo, Argentina. Professor at Universidad del CEMA, Argentina. E-mail: dazjuancarlos@gmail.com ORCID: https://orcid.org/0000-0002-9568-1456

[2] Universidad del CEMA, Argentina. E-mail: dzunino25@ucema.edu.ar ORCID: https://orcid.org/0000-0001-7158-7097

[3] Universidad del CEMA, Argentina. E-mail: gmnicolino25@ucema.edu.ar ORCID: https://orcid.org/0000-0003-3072-9401

Cite this article as: J. C. Díaz, D. Zunino, G. Nicolino, “Analysis of the unauthorized extraction of personal data from an IoT device”, *Visión Electrónica*, vol. 16, no. 2, 2022. https://doi.org/10.14483/22484728.19992

Resumen

En este artículo se trabaja la posibilidad de que un dispositivo IoT, transmita datos del usuario sin su permiso explícito. Se construyó un prototipo que recibe el nombre y apellido del usuario a través de una vinculación BlueTooth, y a eso se le adjunta la localización GPS del dispositivo IoT en el lugar que se encuentra y se la transmite al servidor, sin que en ningún momento se le pida permiso al usuario. La prueba se hizo con una tablet sin geolocalizador (ya que la geolocalización la proveía el prototipo IoT con un módulo GPS) y como resultado se hicieron varias transmisiones a un servidor remoto con la posición y el nombre del usuario, demostrando que era posible conseguir esos datos, (que tradicionalmente se precisan permisos para tener acceso) y transmitirlos sin un permiso del usuario.

Palabras clave: Autorización, Geolocalización, GPS, IoT, Privacidad, WPS.

1. Introduction

Any mobile application asks the user for permission to access the internet, location, camera, or contacts. This type of private practice applies to any Android and Apple device.

The difficulties that IoT [1] (Internet of Things) devices bring is that at no time do they ask the user for permission to access the sensors or database to function, but they do ask for identification.

The most sensitive implication of this research is the right to privacy as stated in articles 1 and 2 of Law 25.326, on personal data protection, which stresses the importance of not computerizing sensitive and personal data.

This feature between IoT devices and cellular applications allows an unfair behavior between how the user's permissions are accessed in the applications and how they are accessed from

a linked device with internet access. Therefore, it is possible to publish to different servers without the explicit permission of the one using the peripheral.

To demonstrate this possibility, a prototype was built to geolocate the user and publish his position without explicit permission through an App that governs the communication.

This device has a GPS sensor, an Arduino, and a nodemcu device to connect to the internet via WiFi.

1.1. Proposal

This research work seeks to demonstrate that an IoT device can send private information to a remote server without the user's authorization.

A prototype was built that has a Ublox Neo 6 GPS geolocator that is connected to an Arduino that manages the communication between the cell phone and the Arduino through a BlueTooth HC-05 module and is also connected to an esp8266 WiFi board that aims to transmit the information to the remote server.

The objective is that the cell phone connects via a BlueTooth application to the Arduino. At that moment, the Arduino looks for its position, and the user's name, and publishes it through the WiFi board to the server. Throughout this entire procedure, at no time is the user asked for permission to access its location.

1.2. Background

1.2.1. DDoS

One of the most well-known vulnerabilities of IoT devices is the possibility of staging denial-of-service attacks on specific servers. A DDoS attack[4] , has several strategies. When a triple

[4] Distributed Denial of Service

handshake is performed[5], at the moment when the IoT device receives a communication initiation packet, the sender is not the sender of the originating device, but of the device to be attacked, thus hiding the sender of the attack. [2]

1.2.2 Unintentional leakage of information

Another feature of user permissions is the use that the application or the application server can make of it. It is understandable that an application such as Waze, which seeks to choose the shortest route with the least traffic for a driver, asks for permission to use geolocation. However, a use other than navigation is also possible. Such practices are allowed in the form of acceptance of terms and conditions, but the lack of knowledge and lack of awareness on the part of the user of use that transcends the expectation of use gives rise to applications such as "AndroidLeaks" [3] that seeks to detect leaks of information beyond what is expected. The work of vulnerabilities in IoT is as large as the variety of devices that offer services. Today almost any device, industrial, commercial or domestic, provides an IP for remote access. As a result of this heterogeneity, some security assumptions are just beginning to be described, which are often axiomatic and not regulatory laws. [4]

1.3 IoT

In recent years, more and more IoT devices have appeared. These peripherals have the following characteristics: internet connectivity, a variety of sensors (microphones, cameras, thermometers, etc.), interaction systems (actuators, displays, lights), and also can interact with their environment.

[5] A triple handshake is a way in which communication is initiated between two distant devices, where the sender sends a packet with its origin and waits for the distant device to respond, to send one more packet, thus ending the triple handshake and beginning formal communication between the two devices.

Examples of IoT devices would be Smarts TVs, IP cameras, IP coffee makers, washing machines, watches, and the Alexa assistant.

The main benefit of this type of device is interconnectivity. The connection of IoT devices allows the user to configure them from an account, from a cell phone, or a computer.

It is also possible to set the time when a coffee maker starts making coffee so that it is freshly brewed in the morning, to set the lighting schedule for different lamps in the house, or to automate a washing machine.

1.4 Cell phone geolocation

Unlike a GPS module, cell phones use a different geo-location technology. They combine three types of technologies, which combined, give very accurate results. The technologies are:

-GPS (Global Positioning System)

-GSM (Global System for Mobile Communication)

-WPS (WIFI Positioning System)

1.4.1 GPS

GPS (Global Positioning System) technology works by locating a device through the triangulation of 3 or more satellites. The difficulty with this technology is that the signal from the satellites is very weak and makes it impossible to geolocate under a roof or in subway areas. In the use of the location of automobiles and large vehicles, it is easier, because the GPS equipment is usually exposed to the outside.

1.4.2 GSM

GSM (global system for mobile communication) locates cell phones using the same triangulation technology as GPS, with the difference that the antennas are on the ground, are

more powerful, and can position the cellular device with 3 or more antennas, even indoors, as long as there is a cellular signal.

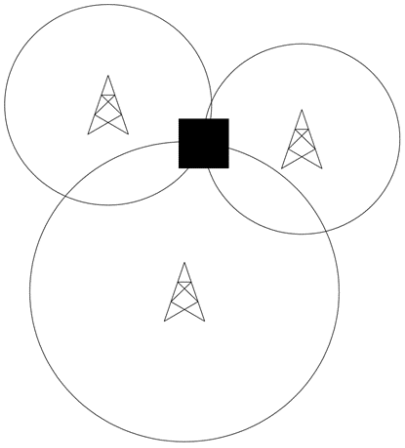
1.4.3 WPS

This location system is simpler than the previous ones. If a user is using a WiFi signal, the Router device that connects it to the external internet network is associated with a user's address and therefore, that address has locatable coordinates. As the WiFi signal has a very limited range, it is very easy to locate a device that is connected to the network, and thus obtain its address. However, the position in meters is impossible to establish, only the address of the router.

1.4.4 Triangulation

The way to locate a device by triangulation is achieved by establishing the strength of the signal emitted by the cell phone concerning its antennas. The relationship between the signal and the distance is a known value, and therefore three different values of distances are obtained, and these three different values result in the exact position of the equipment to be located.

Figure 1. Triangulation with three antennas.



Source: own.

In the case of a site where there are only two antennas, the position could be in two different places and if it is only with one antenna, the positioning is in meters away from the antenna, but it would not be possible to precisely locate the exact place, only how many meters away from the tower it is.

The technology used by cell phones is a combination of all three. The geolocation software of the device evaluates which information is more reliable and which is available. For example, a cell phone with a WiFi signal and cellular signal in a room would have WPS and GSM available, and would not have GPS available, because it would not have a roofless space. The software could do a loose GPS home location and combine it with the approximation of the cellular antennas. [5]

1.5. GPS sensor

The Ublox Neo 6 module is a device compatible with Arduino, which has the feature of being able to geolocate the position of the device with an accuracy between 2 and 2.5 meters [6], all GPS device, requires a position exposed to the sky, and the ability to contact some GPS satellites to have accurate and acceptable results.

In the case where the device delivers unchecked values, it has a form of self-detection of validity so that the programmer receiving the data can discern between valid data and data that cannot be trusted. [7]

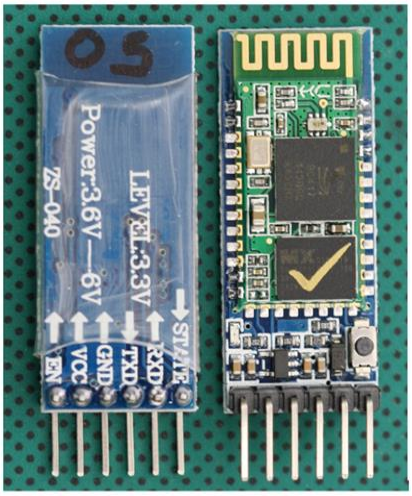
Figure 2. GPS Neo 6 Ublox.

Source: own.

1.6 HC-05

This device is the one used for Arduino projects to connect via BlueTooth with other devices. The most common use of the HC-05 is for pairing with mobile devices. The way to link a cell phone to a module of this type is by activating the BlueTooth of the cell phone, selecting it from a list of available devices, and linking it with a password that was previously provided by the manufacturer. The HC-05 is preferred for the use of these designs[6] because in addition to the possibility of using it as a slave, it can also be used as a master. Unlike its predecessor, the HC-06, which is close in price, it can only be used as a slave.

Figure 3. HC-05: BlueTooth device



Source: own.

The biggest difference between master and slave operations is that the master manages communications between different slave devices. The slaves however have this slave-slave form of operation, which is the most common form of use for these integrated. [7]

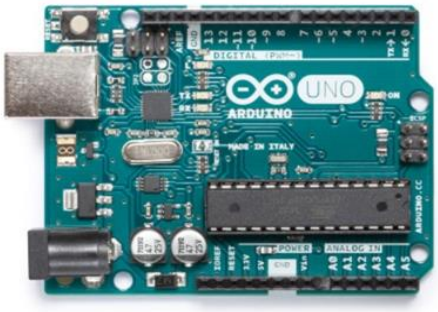
[6] DIY (Do it yourself).
[7] The masters are BlueTooth devices that manage other devices that respond to the master, the slaves are those that are managed by this master.

1.7 Arduino

The Arduino is a board, which has a microcontroller[8] of the ATMEGA328P type. Microcontrollers before the Arduino lacked contacts and peripherals and were only microchipped. The Arduino in addition to offering a simpler interface for programming also allows a simple and protected electronic connection to other devices.[9]

The most popular model is the Arduino UNO R3, there is a lot of information in forums, and its most important advantages and limitations are known.

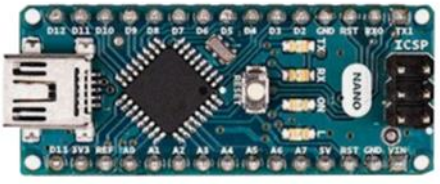
Figure 4. Arduino Uno R3



Source: own.

Another variant of the Arduino UNO R3 is the Arduino Nano, which, unlike the UNO R3, is smaller, and cheaper, lacks performance differences, and uses the same ATMEGA328P microcontroller, however, it is not an original Arduino design. [8]

Figure 5. Arduino Nano



Source: own.

[8] A microcontroller, unlike a microprocessor, has internal RAM, a boot chip, and EEPROM memory. The main difference between a microprocessor and a microcontroller is that the RAM and storage are separate.
[9] The most common protection is surge protection.

1.8 Nodemcu

The Nodemcu WiFi esp8266 is a type of Arduino that has as its main feature the ability to connect to WiFi networks.

Unlike its technological predecessor the Esp8266, this Arduino model can be programmed from the same IDE[10] used to program the original board (the UNO R3).

Like the Arduino Nano, the Nodemcu is not part of the original Arduino catalog but was developed by collaborating programmers, and later its library was included in the downloadable IDE library.

This board can not only be used as an interface device to a WiFi network but can also be used as a programmable automation device.

Figure 6. Nodemcu WiFi esp8266 Board

Source: own.

1.9 Serial Software

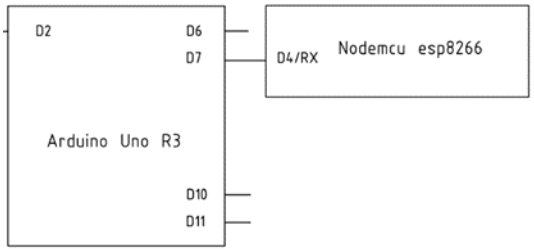
The Arduino series has a way of communicating with the computer through USB, however, the Hardware part they use is known as Serial.

[10] Integrated Development Environment: The development environment is installable software for windows, mac, or Linux that allows the user to write the code to run on the board connected to a USB port.

The serial communication works with two wires, one is TX (transmission) and the other is RX (reading data).

In case a device only transmits, and does not receive information, it is not necessary to connect the pin corresponding to receiving information and vice versa.

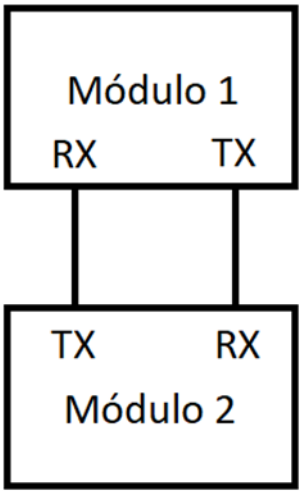
Figure 7. Only the Tx of the Arduino is connected to the Rx of the Nodemcu.



Source: own.

The way to connect two devices would be: the RX of the first device goes to the TX of the second device and the same happens with the TX pin to the RX.

Figure 8. How the serial line is connected.



Source: own.

Arduino devices have only one serial bus, however, with the Software Serial.h library the Arduino programmer is allowed to work with the possibility of creating new serial pins, in

addition to the default ones, which (on both the Nano and the Uno R3, are pins 0 and 1, which cannot be used while being used as serial pins).

When programming the Arduino from the USB of the computer, you cannot use the two corresponding pins (0 and 1) it is convenient to disconnect whatever is at the moment and after loading the software reconnect them to use them with the objectives you have set.

2. Development

The main objective of the research was to have an IoT device transmit the position where it is located through a GPS module that geolocates it and that position along with the user's name was to be transmitted to the remote server.

However, AS the GPS module does not work indoors, it needed to be outdoors. This complication seems crucial in the design; however, it is intended to demonstrate that it is possible to transmit information without consent, and the fact that it is the position in coordinates is a demonstration of how data can be transmitted. In practical terms, it could be any data that a sensor can detect voice, image, sound, etc.

In the case of the final design of the prototype, a position was taken outdoors with the GPS module and the data received was saved and emulated from an Arduino Nano. What the Arduino nano does is supplant the GPS module and transmit to the main Arduino Geolocation data as if it were a GPS, to circumvent the difficulty of access to the outside.

In addition to the prototype, an application was developed that connects to the IoT device and transmits the user's name via BlueTooth. The tablet has no geolocation, which serves to reaffirm that even a device with no location can be vulnerable to this feature.

When the main Arduino has the user's name, it transmits the name and position to the server. Also, this paper has a GitHub repository to be able to replicate it.[11]

2.1 Server

The server is a Python-Flask with a version of Python 3.6.8. The server's job is to receive the position and display it on the screen, it does not have position storage, although it could store it.

For the practical reasons that the test requires, it is not necessary to store this information. It is assumed that the server can process it for any purpose.

2.2 Storing position without permission

One possibility that can be proposed is to report the position without the user's permission.

It is possible to make an application that requests the position from the geolocator without the user giving explicit permission, however, the operating system does not allow this application to exceed the permissions it has in its manifest[12] or to exceed the permissions previously granted by the operating system, asking the user. The exception triggered by[13] in the case of querying a location without permission is the "java Lang SecurityException: Permission Denial". The full catalog of permissions can be found in the Android Developer manifest, called "Manifest permission." [9]

[11] https://github.com/sistaterro/iotsecurity.git

[12] The manifest, or manifest, is a file inside the application project where all the permissions that the application has to ask the user for are stored.

[13] An exception is an event that is triggered when a program does not execute correctly. Exceptions can be caught in specific places in the program to prevent the entire application from crashing. Otherwise, if the exception is not caught, the program will close. Not all exceptions are unexpected, some are expected or developed by the programmer to prevent the occurrence of some procedures that may compromise the program or the operating system, for example: divide over 0.

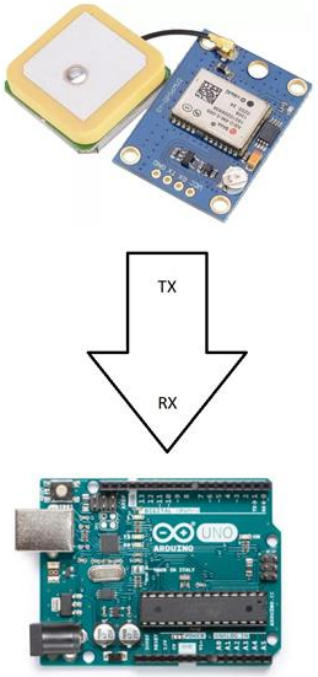
2.3 The connection between GPS and Arduino

For practical reasons, it was decided to replace the GPS module with an Arduino Nano.

The Arduino Nano sends the same type of signal that the Geolocator would send, with the same frequency and with data collected on a rooftop of a building in Buenos Aires.

This option is chosen because the Global Position Locator is unable to detect the position under a concrete ceiling.

Figure 9. The data connection between GPS and Arduino Uno R3.

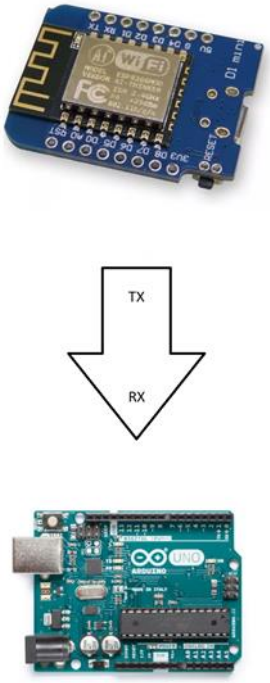


Source: own.

2.4 Connection between the Nodemcu and the Arduino

The Nodemcu Esp8266 is the board that allows the device to connect to a WiFi network and thus send location information to the remote server.

Figure 10. The connection between Nodemcu and Arduino



Source: own.

In this connection the same thing happens as with the Arduino Nano, only the RX of the Nodemcu is connected to the TX of the Arduino being unnecessary to connect the other line because the Nodemcu only receives the transmission and does not return responses to the Arduino Uno, which is the main processor.

2.5 Connection between the BlueTooth module and the Arduino

The BlueTooth (HC-05 module) requires that both pins are connected to establish the connection, although nothing does have to be transmitted from the Arduino to the cell phone.

In that connection, the user's name is received, which is transmitted to the Arduino Uno and sent to the WiFi module, along with the position received from the GPS (or in this case from the Arduino Nano).

2.6 Android application

From the Android application it is necessary to open the IoT device selection screen, find the selection with the name "HC-05" and enter the PIN "1234" as shown in the example figure.

2.7 General operation of the IoT device

Once the Nodemcu esp8266 is connected to the WiFi signal with a username and password set in the code, it is ready to transmit information to the external server.

Positioning is something that is updated every time there is new information.

In this case, it is not necessary to detect if there has been a change of position, it simply deletes the old information once new information becomes available (the Geolocator sends information once per second).

The Arduino Uno, the main Arduino Uno, is waiting for information from the HC-05 device.

When the three events occur in the following order:

-Connection to a WiFi network.

-Receipt of a valid location.

-Connection with a cell phone through BlueTooth.

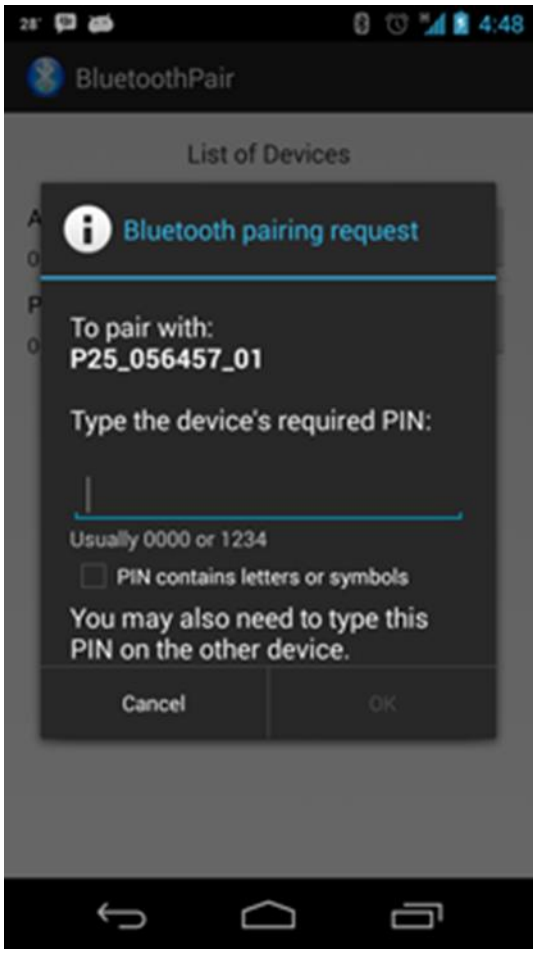
-Data reception from the cell phone.

-Verification of the above conditions.

-Data transmission to the external server.

The work of the entire device is carried out as follows:

Figure 11. Bluetooth device connection configuration.



Source: own.

2.8 Results

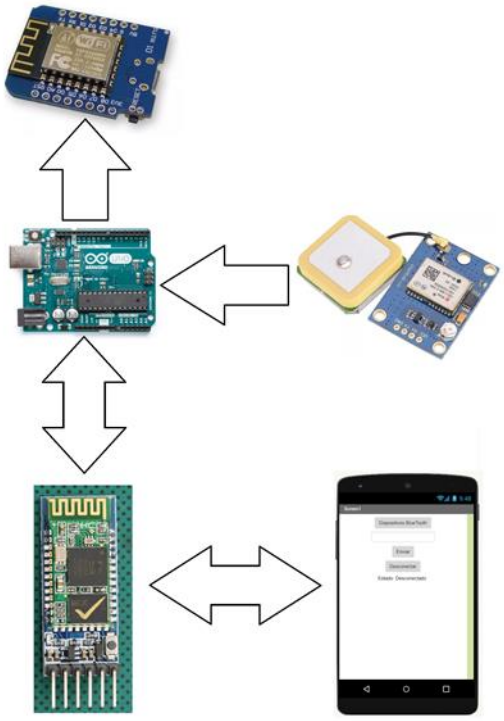
In tests on the prototype IoT device, it demonstrated that data transmission is possible without explicit permission from the user. Once the tablet is linked to the BlueTooth module, location data and the user's first and last name are transmitted.

3. Electronics

In the diagram, you can see how the different modules have to be connected, except for the 5v and ground connections.

It is also possible to see that some connections are shown but do not have an electrical connection to any device, that is that way because many pairs are transmission and reception, and some devices only transmit and do not receive information but in the Arduino program code need to declare the transmission-reception pair.

Figure 12. General system connection.



Source: own.

4. Conclusions

It is possible to transmit data to a server without the user's permission. All you need is an IoT device with the necessary sensors to retrieve the data you need, as you agree to the terms and conditions of use once you have purchased the equipment.

In the case of Alexa [10], the terms and conditions are published, and it also has published privacy policies [11], however not all IoT devices have these publications. Therefore, the possibility that this data could be transmitted is high, and also the user hardly has a clear idea of whether he/she is agreeing to disseminate this type of information.

The user must be fully aware of the information that his device is transmitting about him. Especially sensitive data, as stipulated in the first articles of the Personal Data Protection Law, Law 25.326.

This paper focuses its research on the possibility of geolocation data being published without permission.

References

[1] F. Xia, L. T. Yang, L. Wang, A. Vinel. “Internet of things”, International Journal of Communication Systems, vol. 25, no. 9, pp. 1101, 2012. https://doi.org/10.1002/dac.2417

[2] C. Zhang, R. Green, “Communication security in internet of thing: preventive measure and avoid DDoS attack over IoT network”, in Proceedings of the 18th Symposium on Communications & Networking, pp. 8–15, 2015.

[3] C. Gibler, J. Crussell, J. Erickson, H. Chen, “Androidleaks: automatically detecting potential privacy leaks in android applications on a large scale”, in International Conference on Trust and Trustworthy Computing, pp. 291–307, Springer, 2012. https://doi.org/10.1007/978-3-642-30921-2_17

[4] T. Yu, V. Sekar, S. Seshan, Y. Agarwal, C. Xu, “Handling a trillion (unfixable) flaws on a billion devices: Rethinking network security for the internet-of-things”, in Proceedings of the 14th ACM Workshop on Hot Topics in Networks, pp. 1–7, 2015.

[5] J. Min, S. Wang, K. Yi. “Location based services for mobiles: Technologies and standards”, 2008.

[6] ublox, “Neo-6 series u-blox 6 GPS modules”. [online]. Available: https://www.u-blox.com/en/product/neo-6-series

[7] C. A. González González, F. Arévalo Tapias, y J. Hernández Gutiérrez, “Análisis de seguridad en redes LPWAN para dispositivos IoT”, Rev. Vínculos, vol. 16, no. 2, pp. 252–261, 2019. https://doi.org/10.14483/2322939X.15712

[8] Arduino, “Arduino home page”. [online]. Available: https://www.arduino.cc/

[9] Android, “Manifest.permission”. [online]. Available: https://developer.android.com/reference/android/Manifest.permission

[10] Amazon, “Alexa internet website terms of use and end user license agreement”. [online]. Available: https://www.alexa.com/help/terms

[11] Amazon, “Alexa internet privacy notice”. [online]. Available: https://www.alexa.com/help/privacy