Global Journals LaTeX JournalKaleidoscope™
Artificial Intelligence formulated this projection for compatibility purposes from the original article published at Global Journals. However, this technology is currently in beta. *Therefore, kindly ignore odd layouts, missed formulae, text, tables, or figures.*

Bluelock a Tool to Prevent Bluetooth Attacks

Luz Adriana Peña Salazar

Received: 28 November 2021 Accepted: 19 December 2021 Published: 29 December 2021

Abstract

The mobile device manufacturers and Internet Of Things are searching for more compatible and easy to connect protocols that increase coverage, which generates the user an effective experience when using different devices. Nevertheless, the constant updating process opens a security gap that exposes the users' personal information. Bluetooth is a communication protocol openly used by manufacturers because of their excellent information transfer capacities, allowing hackers to exploit it. The manufacturers work daily, generating security patches for communication protocols but this hasn't been enough to mitigate vulnerabilities. The security has always been on the manufacturer's side but the final user is limited to use a few security customization options to protect his network perimeter. Based on the wireless devices' background we wonder, Is there any way in which the user has the chance to reduce the possible dangers that wireless devices face? We can answer this question through the development of a research in the more vulnerable areas, simulating the attacks on a mobile device to generate a possible solution that allows the user to have more control over his bluetooth connections.

Index terms— bluetooth, android, security, attacks.

1 Introduction

dvances in wireless communication have faced compatibility issues, which has forced technology manufacturers to regulate communication protocols. One of the most popular has been Bluetooth used to interconnect mobile devices and IoT technology. This protocol was created in 1994 by the electric engineer Jaap Haartsen and it has been updated throughout the years that has allowed it to be one the best protocols due to its information transfer rate.

Because of its popularity, coming from manufacturers, this has been the target of hackers who have used vulnerabilities to attack other users and obtained their private information from their mobile devices. The response to these attacks is usually reactive. Such attacks have compromised thousands of users' information, even installing apps that act like malware, that allows access to the peripherals connected to the victim's device, like a camera, mic, GPS, and others.

Smartphones are the most used devices for Bluetooth technology due to its features and the storage capacity in Android. The manufacturers keep reactive during the attacks, generating patches and updates, saving the security recommendations.

To generate a preventive solution, we propose the user is able to manage the Bluetooth interface so it can be used without exposing private information when the device is active. To accomplish this, a testing environment is set up to deploy Bluetooth attacks to identify possible vulnerabilities that could be reduced by an Android app for Smartphones.

2 a) Glossary

Android is a mobile operating system based on a modified version of the Linux kernel and other open source software, designed primarily for touchscreen mobile devices such as smartphones and tablets. Android is developed by a consortium of developers known as the Open Handset Alliance and commercially sponsored by Google.

? Attacks: In this project an attack is a cyberattack, and it is the attempt caused by someone who wants to get control over an informatic system once he gets access to it. The attacks have several purposes in order to cause damage through espionage to get money or to find vulnerabilities in the system. ? BlueLock: It is the created tool for Android devices that allows control over the Bluetooth interface by the user to manage connectivity among active devices. ? Bluetooth: It is a short-range wireless technology standard that is used for exchanging data between fixed and mobile devices over short distances using UHF radio waves in the ISM bands, from We use the Bluelog tool from Kali Linux that allows us to see and register detectable devices on the Bluetooth network and it is executed as you will find below: **??**———————— Blue Scanner is another Kai Linux tool that allows us to scan through the Bluetooth network adapter, capturing connectivity features from connected devices on the network and it is executed as follows root@kali:~# btscanner Keyword "i" produces and scans all devices on the network and generates a file directory called by the MAC's device name with all the information obtained, using these tools we can start to validate the attacks and check their functionality. used interchangeably when referring to internal cyber security tests, but they're not exactly the same. Penetration testing is a type of security test in which an organisation hires a certified professional to assess the strength of its cyber security defences. The goal of ethical hacking -like criminal hacking -is to find security vulnerabilities in an organisation's systems. However, as the word 'ethical' suggests, the person conducting the attack must have the organisation's approval before proceeding. ? Information security: The state of being protected against the unauthorized use of information, especially electronic data, or the measures taken to achieve this.root@kali:~# bluelog Bluelog (v1.1.2) by MS3FGX -

3 Global Journal of Researches in Engineering

? Vulnerability: It is a failure in the security system in which the user can access a system to manipulate information, app or take control over the system.

4 II.

5 Attack Deployment

Knowing the different types of attacks over Bluetooth, they are possible to duplicate through Pentesting. The tests done are known as "white box" (tandem), because we implement the attack environment using the platform pentesting Kali Linux, keeping track of every tool and designing the hacking tests for Bluetooth.

To begin, through Kali Linux we ensure that Bluetooth is activated using the following commands:

? Bluejacking attack: To make this possible, we need to create a contact user, instead of the name, the message will be written, in the directory folder that will act as the attacker and the contact user is saved. After That, an area with several mobile devices is found and the option "send via Bluetooth" is chosen, and it is sent to the target devices. ? Bluesmack: a line executed directly from the attacker's console device. which deploys several calls in a specific frame of time, causing the Bluetooth denial service in the host or hacked device. The script is: lroot@kali:~# while read r; do l2ping -s 50 84:C7:EA:57:36:D7; done < numscans we created a file called: numscans with a timer from 1 to n, to be used in the script and avoid memory leaking in the attacker's device.

root@kali:~# while read r; do l2ping -s 50 84:C7 We can see the Bluetooth protocol is designed to facilitate device connection, and it is a mandatory approach for IoT communication devices, so through the software we designed we do not want to limit or block the protocol's functionality automatically by a service or app, but we want to give the user this option, designing an app that allows him to see the connectivity events, paired devices, and allows the user the chance to block any device at anytime. The diagram shows the device and the app running, participating in different communications with different devices, the device is in a susceptible environment when it is often attacked via Bluetooth. The device has a Bluetooth interface that communicates with the app (Bluelock) that manages the Bluetooth adapter.

6 III.

7 Design Phase

The app (BlueLock) has an event filter that detects specific changes in the Bluetooth adapter, such as incoming connections, and registering all the information in a database, and the app log.

IV.

8 Development Phase

9 App development

As a solution a mobile app for Android is proposed that allow people to control the Bluetooth interface communication, turn on and off the Bluetooth controller, enable finding new devices, consult paired devices, check the Bluetooth event's logs, block access to multiple devices.

10 App features

1. User's interface to control the Bluetooth device. The state the device is found, is extremely important for the app. Therefore, if the app finds a device and this is active, when establishing a connection, it can be blocked to avoid negative actions to our device from this unwanted device. The following method blockDevice validates the state from this device and updates the blocked column from the Devices chart.

11 i. Bluejacking attack results

After executing the test, we can see the app blocks the contact sent, blocking all the communication channels between those devices.

12 b) Bluesnarfing attack test i. Deploying attack

This attack is similar to the previous one, but it enters and steals information from the attacked device. We execute the same previous commands and we get the results shown in the console. root@kali:~# sdptool browse –tree –l2cap C0:8C:71:84:94:A1 Failed to connect to SDP server on C0:8C:71:84:94:A1: Connection timed out root@kali:~# sdptool browse –tree –l2cap C0:8C:71:84:94:A1 Failed to connect to SDP server on C0:8C:71:84:94:A1: Operation already in progress root@kali:~# sdptool browse –tree –l2cap C0:8C:71:84:94:A1 Failed to connect to SDP server on C0:8C:71:84:94:A1: Connection timed out root@kali:~#

13 ii. Bluesnarfing attack results

The attacked device avoids the description services reading and keeps inaccessible to the attacker device.

14 VI.

15 Analysis Results

When we deployed the attacks and used the app Bluelock, we got a positive result due to the effectiveness when blocking attacker devices trying to deploy their attacks. In the bluejacking, it does not allow incoming notifications from the corrupted contact, and in the Bluesnarfing it was completely stopped since the device tried to start communication.

The app, through the blocking device functionality, offers better security and control of the device Bluetooth connections, because it allows the user to choose the devices that will be able to establish a connection and keep track of the bluetooth connections events. After testing the file transfer and Bluetooth attacks we see in the app the functionality filter for the bluetooth adapter states works as expected, so its functionalities could be expanded, defining protocols for every single state of the Bluetooth adapter, increasing the app's usability and security to protect our information from Bluetooth attacks.

The log registration allows us to access the Bluetooth adapter communication events in a short time range and to detect unusual events that take place in the Bluetooth interface range.

The functionality of the app consists in the device's bluetooth adapter control. In case the interface does not respond, it will need to be reseted to get control again, this action can be done by BlueLock.

An app that allows us to see the Bluetooth interface actions is not easily found, because they happen under a transparent communication cape for the user. So with the app BlueLock we can offer a better control of the use that bluetooth connections represent and the events at the moment of establishing connections.

V.

16 Conclusions

The synergy between hacking and the bluetooth attacks, along with software development, allows them to complement each other, facilitating the design, development, testing and deploying of software solutions that improve security in device communication interface, and allow the implementation of closed mobile communication systems.

Most Bluetooth attacks are done while the device's interface is active, due to the protocol working as a receptor waiting for incoming communications and in that state attacks like Bluesnarfing can take place and steal information, taking advantage of the human factor to reach closeness of the target device.

Software solutions for mobile devices focused on connection validation and data packages can improve Android devices' security in places like apartment buildings where there are a lot of mobile devices active and we have a higher chance to be hacked in such environments. BlueLock is a security app oriented to connectivity events that allows unwanted device detection and offers the chance to block them at any time. This is possible by validating the Bluetooth interface state through Intent filters, that is sensible to the Bluetooth adapter changes.

The robustness of the testing set for BlueLock solution, has given positive security results that allow us to block BLuejacking or Bluesnarfing attacks coming from previously blocked devices. In conclusion, we are allowing the user to be responsible for managing the device he allows or not to connect. [1]

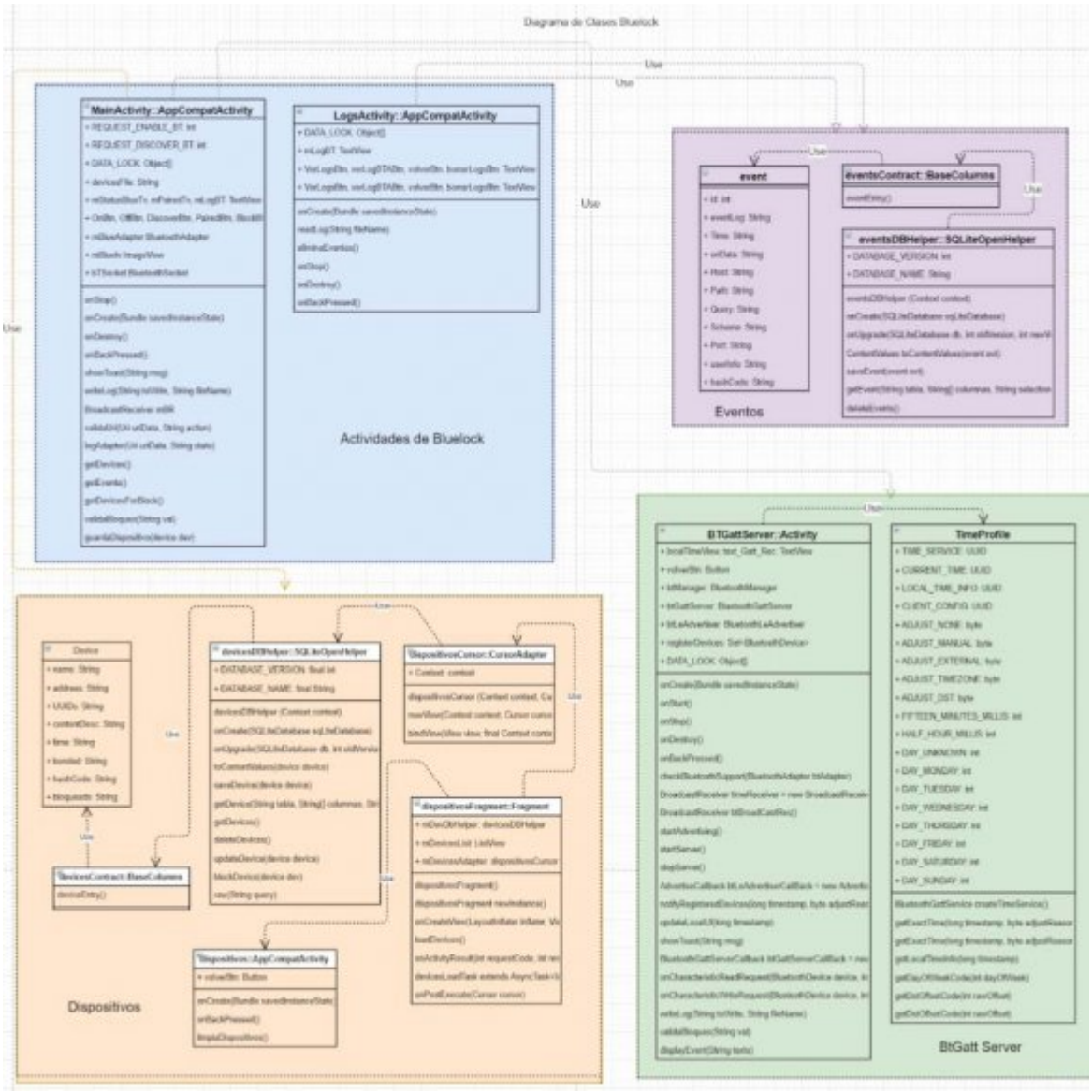
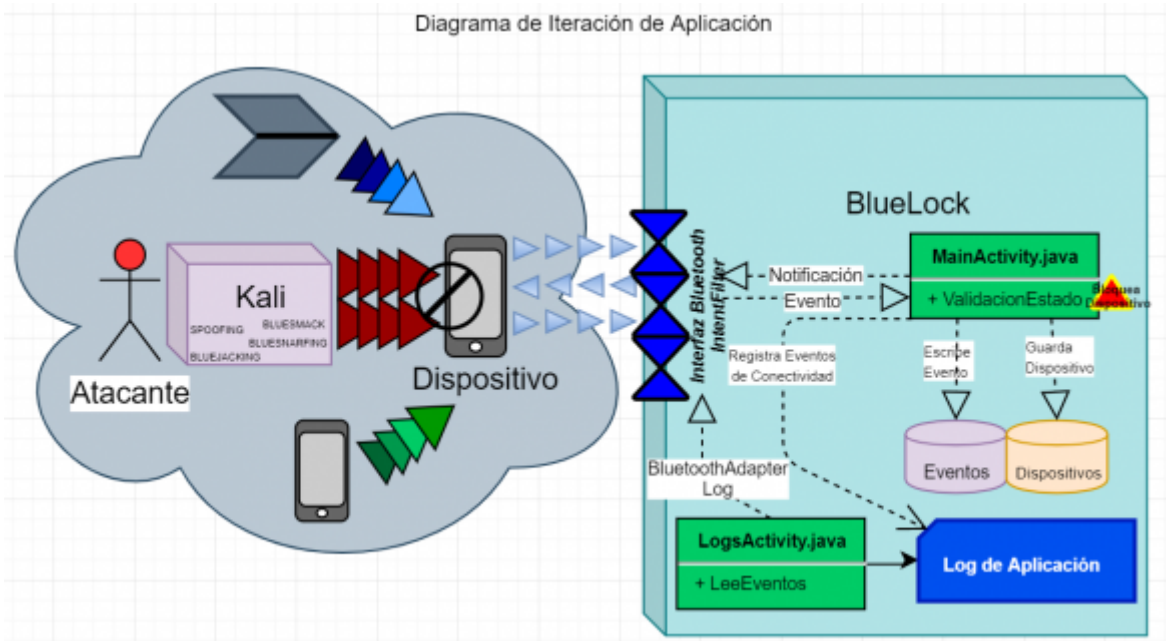


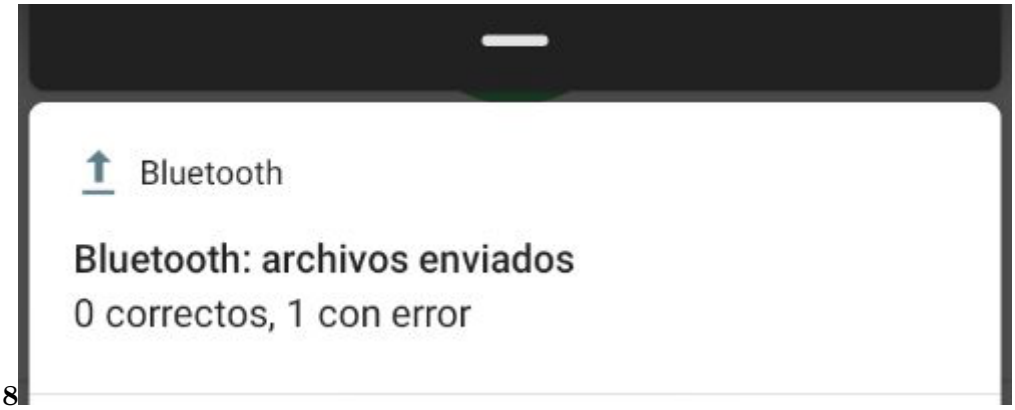
Diagrama de Clases Bluelock
Use
MainActivity::AppCompatActivity
LogsActivity::AppCompatActivity
Actividades de Bluelock
event
eventsContract::BaseColumns
eventsDBHelper::SQLiteOpenHelper
Eventos
BTGattServer::Activity
TimeProfile
BtGatt Server
Device
devicesDBHelper::SQLiteOpenHelper
DispositivosCursor::CursorAdapter
DevicesContract::BaseColumns
DispositivosFragment::Fragment
Dispositivos::AppCompatActivity
Dispositivos

Figure 1:



2

Figure 2: 2 .



8

Figure 3: Fig. 8 :

```
//a1BroadcastReceiver private final BroadcastReceiver mBR = new
BroadcastReceiver() { @Override public void onReceive(Context
context, Intent intent) { final String action = intent.getAction();
String evento = ""; boolean bloqueo = false; final Uri uriData =
intent.getData(); event evt = validaUri(uriData, action); eventsDB-
Helper dbHelper = new eventsDBHelper(getApplicationContext());
String log = "", showToastLog = ""; boolean logger;
if(action.equals(BluetoothAdapter.ACTION_STATE_CHANGED)
||action.equals(BluetoothAdapter.ACTION_DISCOVERY_STARTED)
||action.equals(BluetoothAdapter.ACTION_DISCOVERY_FINISHED)
||action.equals(BluetoothAdapter.ACTION_SCAN_MODE_CHANGED))
{ final int estado = intent.getIntExtra(BluetoothAdapter.EXTRA_STATE,
BluetoothAdapter.ERROR); switch (estado) { case
BluetoothAdapter.STATE_OFF: evento = "<font
color='blue'>BluetoothAdapter.STATE_OFF"; log += lo-
gAdapter(uriData,evento); break; case BluetoothAdapter.STATE_ON:
evento = "<font color='blue'>BluetoothAdapter.STATE_ON";
log += logAdapter(uriData,evento); break; case Blue-
toothAdapter.STATE_TURNING_OFF: evento = "<font
color='purple'>BluetoothAdapter.STATE_TURNING_OFF";
log += logAdapter(uriData,evento); break; case Blue-
toothAdapter.STATE_TURNING_ON: evento = "<font
color='purple'>BluetoothAdapter.STATE_TURNING_ON";
log += logAdapter(uriData,evento); break; Fig. 3: case
BluetoothAdapter.STATE_CONNECTING: evento = "<font
color='purple'>BluetoothAdapter.STATE_CONNECTING"; log
+= logAdapter(uriData,evento); BluetoothDevice device = in-
tent.getParcelableExtra(BluetoothDevice.EXTRA_DEVICE); bloqueo =
validaBloqueo(device.getAddress()); UUID uuid = UUID.randomUUID();
if(bloqueo){ try { Log.w("ACTION_ACL_CONNECTED","Entró a
removeBond"); log += logAdapter(uriData,evento); showToast("Dispositivo
bloqueado ("+uuid+") : "+device.getAddress()+" : "+ device.getName());
} catch (Exception e) { Log.w("ACTION_ACL_CONNECTED",
e.getMessage()); evento = "<font color='red'>Error de Dispos-
itivo Bloqueado ("+uuid+"): "+device.getAddress()+"; "+de-
vice.getName(); } log += logAdapter(uriData,evento); } break;
Fig. 4: case BluetoothAdapter.STATE_CONNECTED: evento =
"<font color='green'>BluetoothAdapter.STATE_CONNECTED";
log += logAdapter(uriData,evento); break; case Blue-
toothAdapter.STATE_DISCONNECTING: evento = "<font
color='green'>BluetoothAdapter.STATE_DISCONNECTING";
log += logAdapter(uriData,evento); break; case Blue-
toothAdapter.STATE_DISCONNECTED: evento = "<font
color='purple'>BluetoothAdapter.STATE_DISCONNECTED";
log += logAdapter(uriData,evento); break; case Blue-
toothAdapter.SCAN_MODE_CONNECTABLE_DISCOVERABLE: evento
= "<font color='purple'>BluetoothAdapter.SCAN_MODE_CONNECTABLE_DISCOVERABLE";
log += logAdapter(uriData,evento); break; case
BluetoothAdapter.SCAN_MODE_CONNECTABLE: evento = "<font
color='purple'>BluetoothAdapter.SCAN_MODE_CONNECTABLE";
log += logAdapter(uriData,evento); break; case Blue-
toothAdapter.SCAN_MODE_NONE: evento = "<font
color='purple'>BluetoothAdapter.SCAN_MODE_NONE"; log +=
logAdapter(uriData,evento); break; default: break; } logger = writeLog(log,
"BluetoothAdapter.txt"); evt.setEventLog(evento); dbHelper.saveEvent(evt);
} logger = writeLog(log, "BluetoothAdapter.txt"); showToast(showToastLog);
```

[Segunda Edición and México] , Segunda Edición , México . Pearson.

[Clarín ()] *Advierten sobre los peligros de utilizar el Bluetooth de tu celular*, Clarín . https://www.clarin.com/tecnologia/advierten-peligros-utilizar-bluetooth-celular_0__WpofinjL.html 2019. Buenos Aires, Argentina. Recuperado de.

[Weed ()] *Bluetooth IoT Applications: From BLE to Mesh. USA: IoT for all*, M Weed . https://www.iotforall.com/bluetooth-iot-applications/ 2018.

[Haataja et al. ()] *Bluetooth Security Attacks: Comparative analysis, attacks and countermeasures*, K Haataja , K Hypponen , S Pasanen , P Toivanen . 2013. Finlandia: Springer.

[Minar and Tarique ()] 'Bluetooth Security Threats and Solutions: A Survey'. N Minar , M Tarique . *International Journal of Distributed and Parallel Systems* 2012. (IJDPS)

[Mitra ()] 'Conoce android studio. USA: Developers Android'. A Mitra . https://developer.android.com/studio/intro/?hl=es-419 *What is Bluestack Attack*, 2017. 2019. (The Security Buddy. Recuperado de)

[Del Cid et al. ()] Alma Del Cid , R Méndez , F Sandoval . *Investigación. Fundamentos y metodología*, 2011.

[Nolasco ()] *Desarrollo de aplicaciones móviles con Android. Segunda edición. Colombia: Ediciones U, Ra -Ma*, J Nolasco . 2016.

[Android ()] *Documentation, android.bluetooth. USA: Developers Android*, Android . https://developer.android.com/reference/android/bluetooth/package-summary 2019.

[Adn Sureste ()] *Este "error" en el Bluetooth pone en riesgo tu celular*, Adn Sureste . https://www.adnsureste.info/este-es-error-en-el-bluetooth-pone-en-riesgo-tu-celular-2130-h/ 2019.

[Carro ()] 'Esto es lo que puede pasar si tienes el bluetooth del móvil siempre conectado'. G Carro . https://www.revistagq.com/noticias/articulo/bluetooth-movil-siempre-conectado-peligros-hackers *Revista GQ. España* 2019. (Recuperado de)

[García ()] *Hablemos de Spoofing. Hacking Ético*, C García . https://hacking-etico.com/2010/08/26/hablemos-de-spoofing 2010.

[Ramos et al. ()] 'Hacking y seguridad de páginas web'. A Ramos , C Barbero , R Martínez , A García , J González . *Colombia: Ediciones U* 2015.

[Conkin et al. ()] *Principles of Computer Security: CompTIA Security+ and Beyond*, A Conkin , G White , C Cothren , R Davis , D Williams . 2018. USA: McGraw Hill Professional.

[Álvarez ()] *Se descubre una "grave vulnerabilidad" en Bluetooth que deja expuestos los dispositivos a posibles ataques*, R Álvarez . https://www.xataka.com/seguridad/se-descubre-grave-vulnerabilidad-bluetooth-que-deja-expuestos-dispositivos-a-posibles-ataques 2017.

[Ciampa ()] *Security+ Guide to Networks Security Fundamentals. Sexta edición*, M Ciampa . 2018. Boston, USA: Cengage.

[Occupytheweb ()] *The Hacks of Mr. Robot: How to Hack Bluetooth*, Occupytheweb . https://null-byte.wonderhowto.com/how-to/hacks-mr-robot-hack-bluetooth-0163586/ 2016. Los Angeles, California. (Wonder How To. Recuperado de)