

CARLOS III UNIVERSITY OF MADRID  
HIGHER TECHNICAL SCHOOL



Universidad  
Carlos III de Madrid

MASTER IN CYBERSECURITY

MASTER THESIS

---

TARGETED EXERCISER FOR ANDROID MALWARE AND GRAYWARE

**Author:** Mario Herreros Díaz

**Mentors:** Juan E. Tapiador, Guillermo Suarez-Tangil

September, 9th, 2016

## Abstract

Nowadays each person has one or more mobile device (smartphones, tablets, wearables, etc) with similar characteristics and performance of a personal computer. Many people use these devices to check their email or make bank transfers. For this, these devices store a lot of sensitive information that it could be very attractive to an attacker.

Mobile devices share same security troubles can be found in a conventional computer as malware hidden inside apps or users that don't use nor protect their correctly. Malware has evolved and is able to by pass protection systems and works only under certain circumstances to avoid detection. Some kind of malware is activated depends of network connection, device location, apps installed, calls received or other type of events. The purpose of this Master Thesis is to study the behavior of malware on mobile devices depending of the device context, focusing on Android operating system. It has developed a system that allows dynamic and automatic generation of many different contexts to study the behavior of Android malware in each of the different scenarios.

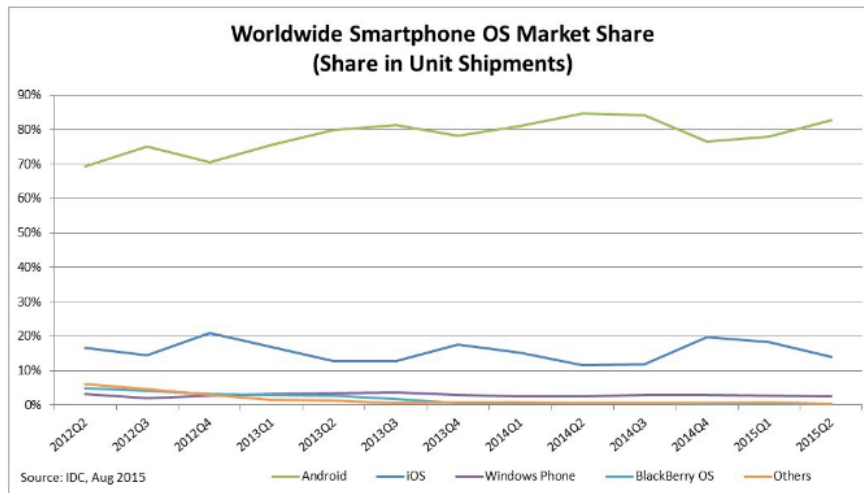
It has been designed a technique based on developing of a new language which defines each of the scenes and events to execute over the device to analyze the malware, in order to study their behavior depending on the characteristics of each context. In this way, it can detect the context features that makes triggering the malicious malware or greyware actions.

This Thesis focuses on analyzing all possible Android events to define a language and developing a system available to understand it and generate automatic executions to launch these events. With this system it is possible detect malware and grayware that a conventional static and dynamic analysis could not detect.

*Keywords: Malware, Greyware, Android, Context, Malware detection, Dynamic analysis, Mobile device, Malware behaviour*

## I. Introduction

Nowadays mobile devices are very present within society and every people uses them in their personal and laboral life. The most popular smartphone OS is Android with a market share between 82 – 84 %, for this reason Android is the platform most targeted by malware and greyware.

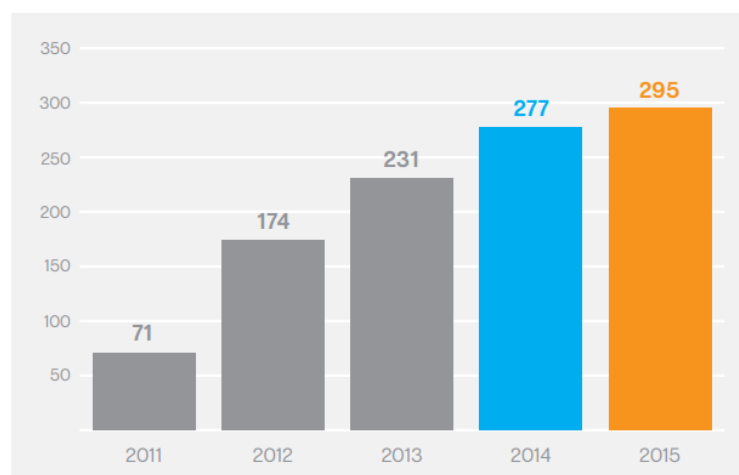


Period	Android	iOS	Windows Phone	BlackBerry OS	Others
2015Q2	82.8%	13.9%	2.6%	0.3%	0.4%
2014Q2	84.8%	11.6%	2.5%	0.5%	0.7%
2013Q2	79.8%	12.9%	3.4%	2.8%	1.2%
2012Q2	69.3%	16.6%	3.1%	4.9%	6.1%

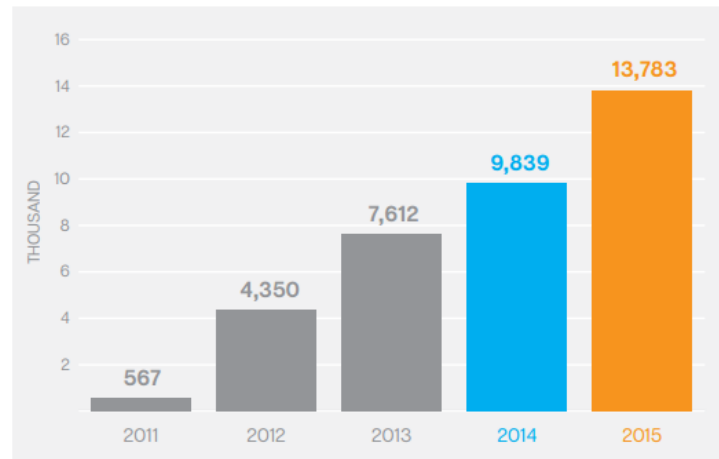
Source: IDC, Aug 2015

<http://www.idc.com/prodserv/smartphone-os-market-share.jsp>

The amount of Android malware families has been incremented very significantly in recent years, it being categorized 295 different families



An important fact is the variability of the Android malware. There are thousand types of malware although, the target functionality is similar. During the last years, the variability of malware has been risen considerably, reaching in 2015 the number of 13,783 thousand types.



The powerful capabilities of mobile devices in networking, computing and sensing has increased the complexity, power and variability of malware.

This allows the malware can evolve and adopt very specific functionality. There is a type of malware that focuses on a specific victims, known as malware targeted. This type of malware focuses on user behavior using his smartphone and other factors relating to him, such as his location, the information handled or applications installed in the device.

The main advantage of targeted malware against security systems is their difficult detection. This is due to the fact that only they execute his malicious code if a concrete context is fulfilled. This situation causes the need to use a lot of resources to detect this type of malware.

In the application stores a large number of applications that must be analyzed before publication are received each day. This causes malware analysis are not made sufficiently comprehensive to detect targeted malware.

This problem can be alleviated by using dynamic analysis techniques that allow the generation of different contexts and behaviors in an agile and automatic way.

#### A. Motivation

Deficiencies in malware detection systems and strong growth of the power and complexity of malware has caused the existence of a major security threat on mobile devices. The analysis of targeted malware is a costly task, so the development of a system that will facilitate and automate the analysis can alleviate this problem existing in mobile security nowadays.

Focusing on Android system, which is one of the most affected OS by mobile malware, it has developed a system able to define and execute different behaviors and contexts, in an effective manner using a virtual sandbox.

### *B. Main contributions*

This work provides the following contributions:

- 1) An analysis and study of Android system and architecture for inject different kind of events and context.
- 2) An analysis and study of targeted malware, focusing on key behaviors and contexts that used to trigger the malware.
- 3) Generation of a new language to easily define a set of behaviors based on different scenarios and contexts.
- 4) An analysis and developing of a new system, based in Targetdroid, capable of computing the new defined language and generate a dynamic analysis based on the contexts and behaviors.

### *C. Structure of the document*

The remainder of this document is structured as follows. First, section II presents the related work regarding Android targeted malware and Targetdroid.

Finally, Section III presents a series of recommendations and discusses about the importance of the security measures in Android malware and other Smartphone systems, and Section IV concludes the work.

Additionally, two appedix...

## **II. Background and related work**

The precursor knowledge of this thesis is the investigation explained in Detecting Targeted Smartphone Malware with Behavior-Triggering Stochastic Models. This investigation focuses on the need of detect targeted malware in smartphones. For this, it has developed a system called Targetdroid that transforms different inputs in a dynamic analysis using multiple behavioral models.

This system uses models that perform different emulations of user behaviors in order to detect the context makes activate the targeted malware or greyware. This system was based on Stochastic Models Triggering as Markov.

The principal problem is reproduce an appropriate set of conditions that will trigger the malicious behaviour. It is necessary execute all possible states and combinations in the worst case. In this paper, it has proposed a novel system for mining the behaviour of apps in different user-specific contexts and usage scenarios.

The target of this thesis focuses on evolving this system to accept flexible inputs and achieve a better definition of the user behavioral.

Other authors also studied the problema of detecting targeted malware such as PyTrigger, focusing on Personal Computers (PC), or CupperDroid that generates different behavioral profiles automatically to analyze malware.

### **III. TargetDroid**

This section will define the system developed that extends the functionality of Targetdroid accepting the new model defined based on the Behavioural User Language

#### **Behavioural User Language**

It has designed a language that allows the definition of user behavior using a hierarchy of scenarios, contexts and events.

This language is based on JSON text format that serving a particular structures and it is defined by the following json schema (link to json schema).

This language will be used for generating user behavior files, used as input and interpreted by the system developed to generate different dynamic analysis on a cloud (for this case an Android emulator).

The language is structured in the following artifacts:

- Scenario: it is a set of contexts.
- Context: it is a set of events.
- Event: atomic action performed on the cloud system.

#### ***Scenario***

This artifact represents all possible wanted scenarios to simulate a specific user behavior. The definition of scenarios allows performing analysis in wich you want to maintain context configurations.

#### ***Context***

This artifact describes all possible events that can be performed during the user behavior simulation determined by a moment in time.

There are three types of different contexts:

- Emulator configuration context (t = -1)  
It defines all environment properties used by the following contexts, that is to say that presents the characteristics of the emulator to be used as a sandbox for the dynamic analysis.  
Engloba todos aquellos eventos que van a ser ejecutar en el momento igual a -1 (utilizando una linea temporal como representante de la ejecución).
- OS configuration context (t = 0)  
In this context assigns the initial state of the initialized system, in other words, configure the emulated Android system at level of OS.  
Engloba todos aquellos eventos que van a ser ejecutados en el momento igual a 0, establece un estado inicial al sistema en el que se van a realizar la inyección de contextos de ejecución.

- Execution context ( $t > 0$ )

It represents each of the events that correspond to user manipulation with the device, such as sending a message, receive a call, run an app or determinate a geolocation.

Engloba todos aquellos eventos que van a ser ejecutados en momentos superiores a 0.

### **Event**

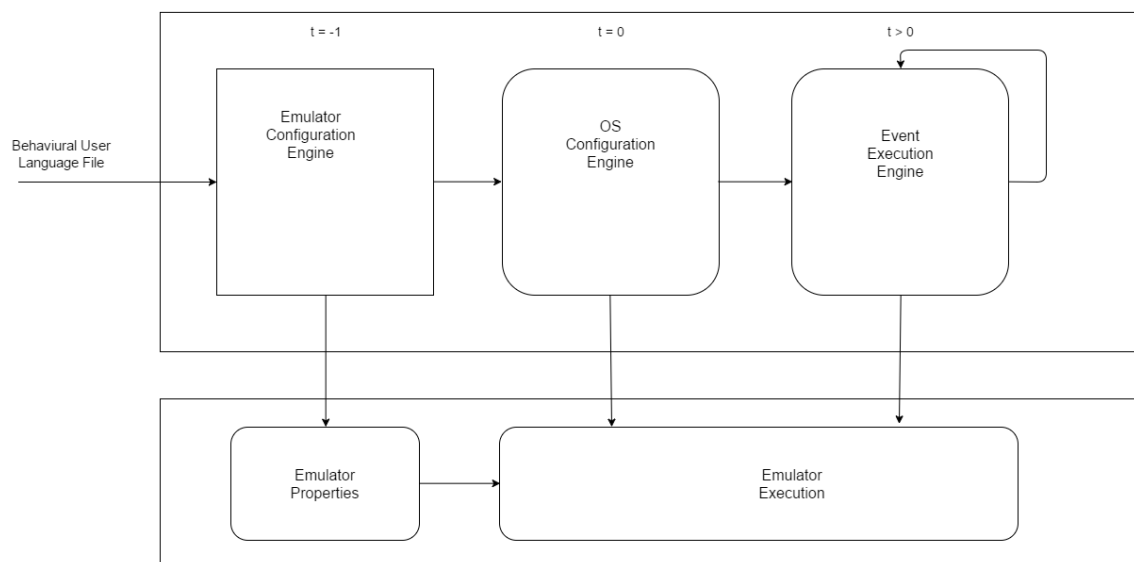
This artifact defines each of the possible actions to be performed on the system to simulate the user behavior.

It is the atomic unit in this system and includes a very different set of artifacts, such as hardware configurations or Telnet calls to emulator that allow simulate a phone call.

### **Architecture**

This section describes the architecture of the system developed. It will explain the components and their functionality inside the system.

The next Figure describes an architecture diagram of the system developed:



It divides the system in two principal components divided in several subcomponents:

- TagetDroid
  - Emulator Configuration Engine:  
This component generates the emulator configuration before running.
  - OS Configuration Engine:  
This component generates the emulator configuration after running.
  - Event Execution Engine:  
This component generates all active user behavior events.
- Emulator
  - Emulator Properties:

This component represents all the configuration properties to define the emulator used such as cloud before running.

- This component represents the emulator in running state, ready to listen commands from the Targetdroid system to execute events.

### **Targetdroid**

The principal components of Targetdroid are composed by different functional modules:

- Emulator Configuration Engine (t = -1):  
There are two principal modules in this component:
  - Android emulator properties file:  
This module parse the emulator properties
  - Command line options  
There is a set of additional options to create and run the emulator by command line, such as define the sdk version, some networks options, etc.
- OS Configuration Engine (t = 0):  
It is formed by two modules:
  - Install app module:  
This module allows install different android apps using adb.
  - OS configuration event injection:  
This module can send events to andoird emulator to configure the initial state of the emulator. It can allow configurate the next options:
    - Timezone
    - Power properties
- Event Execution Engine (t > 0)  
It is composed by the following modules:
  - Intents:  
This module parses a set of adb commands related to execute intents to generate specific events in the emulator: (table)
    - Start intent
    - Start service intent
    - Broadcast intent
  - ADB command to event:  
This module parses a set of adb commands to generate a common user events in the emulator: (table)
    - Connect wifi:
    - Lock screen:
    - Unlock screen:
    - Volume up
    - Go home:
    - Take screenshot:
    - Start clock app
    - Stop clock app
    - Start wifi
    - Check wifi status
    - Enable wifi
    - Disable wifi



- Enable mobile data
- Disable mobile data
- Telnet command to event:  
This module uses the Telnet protocol to send commands to emulator and generate a set of events in the cloud. The events related to telnet commands are the following:
  - GSM events
  - Power events
  - Call events
  - SMS events
  - GEO events
- Monkey tester:  
This module allows run monkey scripts, defined previously, such as event to generate UI events during the analysis. This module uses the android tool adb to generate these kind of events.

### **Emulator**

For this system, it has used the Android SDK mobile device emulator. It lets generate different virtual mobile devices with many configurations and interact with them using different tools, such as adb, telnet, avd manager or logcat.

The principal modules used in this system are the following:

- Properties.ini file:  
Every AVD (Android Virtual Device) created has an unique properties file. This file describes the properties of a given virtual device configuration file. It allows configure a set of characteristics for the virtual mobile device created.
- Command line options  
When it runs an AVD by command line, there are different options to configure the emulator, such as network or sytem options.
- AVD Manager  
It is a tool to create and configure Android Virtual Device. This tool is used to generate every cloud with the properties defined in each scenario.
- ADB  
Android Debug Bridge is a versatile command line tool that lets you communicate with an emulator instance or connected Android-powered device. It is a client-server program that includes three components:
  - A client, which sends commands. The client runs on your development machine. You can invoke a client from a shell by issuing an adb command. Other Android tools such as DDMS also create adb clients.
  - A daemon, which runs commands on a device. The daemon runs as a background process on each emulator or device instance.

- A server, which manages communication between the client and the daemon.

The server runs as a background process on your development machine.

- Telnet

Telnet is an application layer protocol used on the Internet or local area networks to provide a bidirectional interactive text-oriented communication facility using a virtual terminal connection.

It is possible to connect to an Android emulator using this protocol and send commands to control the emulator.

- IV. Discussion  
TODO
- V. Conclusion  
TODO
- VI. Future Works  
TODO

## References

TODO

## Appendix A : json-schema

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "id": "input JSON to configure a scenario and launch android events",
  "type": "object",
  "required": ["name", "time"]
  "properties": {
    "name": {
      "id": "scenario name",
      "type": "string"
    },
    "time": {
      "id": "definition of context in every execution time",
      "type": "object",
      "oneOf": [
        {
          "type": "object",
          "required": ["id"],
          "properties": {
            "id": {
              "id": "execution time",
              "type": "integer",
              "enum": [-1]
            },
            "emulator": {
```

```
"id": "emulator configuration params",
"properties": {
  "name": {
    "id": "emulator name",
    "type": "string"
  },
  "port": {
    "id": "emulator port",
    "type": "integer"
  },
  "sdk_version": {
    "id": "emulator android version",
    "type": "string",
    "enum":["android-8", "android-10", "android-15", "android-16", "android-17",
"android-18", "android-19", "android-20", "android-21", "android-22", "android-23"]
  }
},
"phone": {
  "id": "phone properties",
  "type": "object",
  "properties": {
    "brand": {
      "id": "phone brand name",
      "type": "string"
    },
    "device": {
      "id": "phone device name",
      "type": "string"
    },
    "hardware": {
```

```
    "id": "phone hardware name",
    "type": "string"
  },
  "imei": {
    "id": "phone imei",
    "type": "string"
  },
  "imsi": {
    "id": "phone imsi",
    "type": "string"
  },
  "network_provider": {
    "id": "phone network provider name",
    "type": "string"
  }
},
"network": {
  "id": "network properties",
  "type": "object",
  "properties": {
    "IP": {
      "id": "phone IP",
      "type": "string"
    },
    "DNS": {
      "id": "phone DNS server",
      "type": "string"
    }
  }
},
}
```

```
"ini_properties":{
  "id": "properties to configurate the emulator",
  "type": "object",
  "properties": {
    "cpu_arch": {
      "id": "The CPU Architecture to emulator",
      "type": "string"
    },
    "cpu_model": {
      "id": "The CPU model (QEMU-specific string)",
      "type": "string"
    },
    "ram_size": {
      "id": "Device ram size",
      "type": "integer"
    },
    "screen_type": {
      "id": "Defines type of the screen",
      "type": "string"
    },
    "main_keys": {
      "id": "Hardware Back/Home keys",
      "type": "boolean"
    },
    "track_ball": {
      "id": "Track-ball support",
      "type": "boolean"
    },
    "keyboard": {
      "id": "Whether the device has a QWERTY keyboard",
      "type": "boolean"
    }
  }
}
```

```
,
"keyboard_lid": {
  "id": "Whether the QWERTY keyboard can be opened/closed",
  "type": "boolean"
},
"keyboard_charmap": {
  "id": "Name of the system keyboard charmap file",
  "type": "string"
},
"d_pad": {
  "id": "Whether the device has DPad keys",
  "type": "boolean"
},
"gsm_modem": {
  "id": "phone DNS server",
  "type": "string"
},
"gps": {
  "id": "Whether there is a GPS in the device",
  "type": "boolean"
},
"battery": {
  "id": "Whether the device can run on a battery",
  "type": "boolean"
},
"accelerometer": {
  "id": "Whether there is an accelerometer in the device",
  "type": "boolean"
},
"audio_input": {
  "id": "Whether the device can record audio",
```

```
    "type": "boolean"
  },
  "audio_output": {
    "id": "Whether the device can play audio",
    "type": "boolean"
  },
  "sd_card": {
    "id": "Whether the device supports insertion/removal of virtual SD Cards",
    "type": "boolean"
  },
  "sd_card_path": {
    "id": "SD Card image path",
    "type": "string"
  },
  "cache": {
    "id": "Whether we use a /cache partition on the device",
    "type": "boolean"
  },
  "cache_path": {
    "id": "Cache partition to use on the device",
    "type": "boolean"
  },
  "cache_size": {
    "id": "Cache partition size",
    "type": "string"
  },
  "lcd_width": {
    "id": "LCD pixel width",
    "type": "integer"
  },
  "lcd_height": {
```



```
    "id": "LCD pixel height",
    "type": "integer"
  },
  "lcd_depth": {
    "id": "LCD color depth",
    "type": "integer"
  },
  "lcd_density": {
    "id": "Abstracted LCD density",
    "type": "integer"
  },
  "lcd_backlight": {
    "id": "LCD backlight",
    "type": "boolean"
  },
  "gpu": {
    "id": "Enable/Disable emulated OpenGL ES GPU",
    "type": "boolean"
  },
  "camera_back": {
    "id": "Configures camera facing back",
    "type": "string"
  },
  "camera_front": {
    "id": "Whether the device can run on a battery",
    "type": "string"
  },
  "heap_size": {
    "id": "Max VM application heap size",
    "type": "integer"
  },
}
```

```
"sensor_proximity": {  
  "id": "Whether there is an proximity in the device",  
  "type": "boolean"  
},  
"sensor_magnetic_field": {  
  "id": "Provides magnetic field sensor values",  
  "type": "boolean"  
},  
"sensor_orientation": {  
  "id": "Provides orientation sensor values",  
  "type": "boolean"  
},  
"sensor_temperature": {  
  "id": "Provides temperature sensor values",  
  "type": "boolean"  
},  
"use_ext4": {  
  "id": "Specifies which file system to use: ext4 of yaffs2",  
  "type": "boolean"  
},  
"kernel_path": {  
  "id": "Path to the kernel image",  
  "type": "string"  
},  
"kernel_parameters": {  
  "id": "kernel boot parameters string",  
  "type": "string"  
},  
"ram_disk_path": {  
  "id": "Path to the ramdisk image",  
  "type": "string"
```

```
,
"system_partition_path": {
  "id": "Path to runtime system partition image",
  "type": "string"
},
"system_partition_init_path": {
  "id": "Initial system partition image",
  "type": "string"
},
"system_partition_size": {
  "id": "Ideal size of system partition",
  "type": "integer"
},
"data_partition_path": {
  "id": "Path to data partition file",
  "type": "string"
},
"data_partition_init_path": {
  "id": "Initial data partition",
  "type": "boolean"
},
"data_partition_size": {
  "id": "Ideal size of data partition",
  "type": "integer"
},
"snap_storage_path": {
  "id": "Path to snapshot storage",
  "type": "string"
}
}
}
```

```
}  
,  
{  
  "type": "object",  
  "required": ["id"],  
  "properties": {  
    "id": {  
      "id": "execution time",  
      "type": "integer",  
      "enum": [0]  
    },  
    "install_app": {  
      "id": "apks to install",  
      "type": "string"  
    },  
    "timezone": {  
      "id": "timezone for the emulator",  
      "type": "string"  
    },  
    "telnet": {  
      "id": "telnet commands",  
      "type": "object",  
      "properties": {  
        "power": {  
          "id": "power properties",  
          "type": "object",  
          "properties": {  
            "ac": {  
              "id": "ac properties",  
              "type": { "enum": ["on", "off"] }  
            }  
          }  
        },  
        "ac": {  
          "id": "ac properties",  
          "type": { "enum": ["on", "off"] }  
        }  
      }  
    }  
  }  
}
```

```

    "capacity": {
      "id": "capacity properties",
      "type": "integer"
    },
    "health": {
      "id": "power health properties",
      "type": {"enum": ["unknown", "good", "overheat", "dead", "overvoltage",
"failure"]} }
  },
  "present": {
    "id": "battery presence",
    "type": "boolean"
  },
  "status": {
    "id": "battery status",
    "type": { "enum": ["unknown", "charging", "discharging", "not-charging",
"full"]} }
}
}
}
}
}
}
},
{
  "type": "object",
  "required": ["id"],
  "properties": {
    "id": {
      "id": "execution time",
      "type": "integer",
      "minimum": 1
    }
  }
}

```

```

    },
    "telnet": {
      "id": "telnet commands",
      "type": "object",
      "properties": {
        "power": {
          "id": "power properties",
          "type": "object",
          "properties": {
            "ac": {
              "id": "ac properties",
              "type": { "enum": ["on", "off"] }
            },
            "capacity": {
              "id": "capacity properties",
              "type": "integer"
            },
            "health": {
              "id": "power health properties",
              "type": { "enum": ["unknown", "good", "overheat", "dead", "overvoltage",
"failure"] }
            },
            "present": {
              "id": "battery presence",
              "type": "boolean"
            },
            "status": {
              "id": "battery status",
              "type": { "enum": ["unknown", "charging", "discharging", "not-charging",
"full"] }
            }
          }
        }
      }
    }
  }
}

```

```
},
"gsm": {
  "id": "calling events",
  "type": "object",
  "properties": {
    "call": {
      "id": "call to a number",
      "type": "string"
    },
    "accept": {
      "id": "accept a call",
      "type": "string"
    },
    "busy": {
      "id": "busy a call",
      "type": "string"
    },
    "cancel": {
      "id": "cancell a call",
      "type": "string"
    },
    "data": {
      "id": "data connection status",
      "type": { "enum": ["unregistered", "home", "roaming", "searching",
"denied", "off", "on"] }
    },
    "voice": {
      "id": "voice connection status",
      "type": { "enum": ["unregistered", "home", "roaming", "searching",
"denied", "off", "on"] }
    }
  }
}
```

```
},
"sms": {
  "id": "send a sms message",
  "type": "object",
  "properties": {
    "number": {
      "id": "number phone to send a sms",
      "type": "string"
    },
    "text": {
      "id": "message",
      "type": "string"
    },
    "required": [ "number", "sms" ]
  }
},
"geo": {
  "id": "location events",
  "type": "object",
  "properties": {
    "fix": {
      "id": "http://jsonschema.net/t1/geo/fix",
      "type": "object",
      "properties": {
        "longitude": {
          "id": "longitude",
          "type": "string"
        },
        "latitude": {
          "id": "latitude",
          "type": "string"
        }
      }
    }
  }
}
```



```
    },
    "altitude": {
        "id": "altitude",
        "type": "string"
    },
    "required": [ "longitude", "latitude" ]
}
},
"nmea": {
    "id": "nmea",
    "type": "string"
}
}
}
}
},
"adb": {
    "id": "adb popular commands",
    "type": "array",
    "properties": {
        "action": {
            "id": "adb action name",
            "type": "string",
            "enum": ["connect_wifi", "lock_screen", "unlock_screen", "volume_up",
"go_home", "take_screenshot", "start_clock_app", "stop_clock_app", "start_wifi",
"wifi_status", "enable_wifi", "disable_wifi", "enable_mobile_data",
"disable_mobile_data"]
        }
    }
},
"intents": {
    "id": "intents commands",
```

```
"type": "array",
"properties": {
  "type": {
    "id": "intent type",
    "type": "string",
    "enum": ["start", "start_service", "broadcast"]
  },
  "params": {
    "id": "intent params",
    "type": "string"
  }
},
"monkey": {
  "id": "monkey script events",
  "type": "object",
  "properties": {
    "path": {
      "id": "path to monkey script",
      "type": "string"
    },
    "package": {
      "id": "apk package where run monkey script",
      "type": "string"
    }
  }
}
}
```

```
}  
  
}
```

## Appendix B: event list

### 1 EVENTOS DE CONFIGURACIÓN DEL ENTORNO

---

Atributo	Valores	Descripción
<b>emulator</b>		
<b>Emuator.name</b>	String	Nombre del AVD a crear
<b>Emulator.port</b>	Integer	Puerto en el que escuchará el AVD
<b>Emulator.sdk_version</b>	String "enum": [ "android- 8", "android- 10", "android- 15", "android- 16", "android- 17", "android- 18", "android- 19", "android- 20", "android- 21", "android- 22", "android- 23"]	Versión del Android SDK utilizado por el emulador
<b>phone</b>		
<b>phone.brand</b>	string	Indica la marca comercial del dispositivo emulado
<b>phone.device</b>	String	Indica el modelo del dispositivo emulado
<b>phone.imei</b>	String	Indica el valor IMEI que tendrá el dispositivo emulado.
<b>Phone.imsi</b>	String	Indica el valor IMSI que tendrá el dispositivo emulado.
<b>Phone.provider</b>	string	Indica el nombre del proveedor de red del emulador
<b>Network</b>		
<b>Network.IP</b>	String	Indica la dirección IP del dispositivo emulado

<b>Network.DNS</b>	string	Indica las direcciones DNS a utilizar por el dispositivo emulado. Puede ser una lista de hasta cuatro direcciones IP o hostnames separado por comas.
<b>ini_properties</b>		
<b>ini_properties.cpu_arch</b>	String	Indica el tipo de la arquitectura de la CPU del emulador.
<b>ini_properties.cpu_model</b>	String	Indica el modelo de la CPU
<b>ini_properties.ram_size</b>	Integer	Indica el tamaño de la memoria RAM
<b>ini_properties.screen_type</b>	String "enum": [ "touch", "multi-touch", "no-touch"]	Define el tipo de pantalla
<b>ini_properties.main_keys</b>	Boolean	Activa/desactiva los botones de Volver/Inicio
<b>ini_properties.keyboard</b>	Boolean	Indica si el emulador utiliza es un teclado QWERTY
<b>ini_properties.keyboard_lid</b>	Boolean	Permite abrir el teclado QWERTY o no
<b>ini_properties.keyboard_charmap</b>	string	Nombre del sistema charmap del teclado
<b>ini_properties.d_pad</b>	Boolean	Indica si el dispositivo presenta teclas DPad
<b>ini_properties.gsm_modem</b>	String	Indica el módem GSM del dispositivo
<b>ini_properties.gps</b>	Boolean	Indica la presencia de GPS en el dispositivo
<b>ini_properties.battery</b>	Boolean	Indica la presencia de batería en el dispositivo
<b>ini_properties.accelerometer</b>	Boolean	Indica la presencia de acelerómetro en el dispositivo
<b>ini_properties.audio_input</b>	Boolean	Indica la presencia de entrada de audio en el dispositivo
<b>ini_properties.audio_output</b>	Boolean	Indica a presencia de salida de audio en el dispositivo
<b>ini_properties.sd_card</b>	Boolean	Indica si el dispositivo soporta el uso de tarjetas SD
<b>ini_properties.sd_card_path</b>	String	Presenta la ruta de la imagen de la tarjeta SD a utilizar
<b>ini_properties.cache</b>	Boolean	Indica el uso de una partición cacheada en el dispositivo
<b>ini_properties.cache_path</b>	String	Presenta la ruta de la cache a utilizar
<b>ini_properties.cache_size</b>	String	Es el valor del tamaño de la caché
<b>ini_properties.lcd_width</b>	Integer	Ancho del pixel LCD
<b>ini_properties.lcd_height</b>	Integer	Alto del pixel LCD
<b>ini_properties.lcd_depth</b>	Integer	Profundidad de color LCD
<b>ini_properties.lcd_density</b>	Integer	Densidad LCD

	"enum": [120, 160, 240, 213, 320]	
<b>ini_properties.lcd_backlight</b>	Boolean	Presencia de LCD Backlight
<b>ini_properties.gpu</b>	Boolean	Indica el uso de una OpenGL ES GPU emulada
<b>ini_properties.camera_back</b>	String	Configuración de la cámara trasera
<b>ini_properties.camera_front</b>	String	Configuración de la cámara frontal
<b>ini_properties.heap_size</b>	Integer	Define el tamaño máximo del heap del emulador
<b>ini_properties.sensor_proximity</b>	Boolean	Presencia de sensor de proximidad
<b>ini_properties.sensor_magnetic_field</b>	Boolean	Presencia de sensor magnético
<b>ini_properties.sensor_orientation</b>	Boolean	Presencia de sensor de orientación
<b>ini_properties.sensor_temperature</b>	Boolean	Presencia de sensor de temperatura
<b>ini_properties.use_ext4</b>	Boolean	Indica que sistema de ficheros utilizar: Ext4/ yaffs2
<b>ini_properties.kernel_path</b>	String	Presenta la ruta de la imagen del kernel a utilizar
<b>ini_properties.kernel_parameters</b>	String	Indica los parámetros a pasar al boot kernel
<b>ini_properties.system_partition_path</b>	string	Indica la ruta de la partición del sistema
<b>ini_properties.system_partition_init_path</b>	String	Indica la ruta de la partición del sistema inicial
<b>ini_properties.system_partition_size</b>	Integer	Tamaño del sistema de partición
<b>ini_properties.data_partition_path</b>	String	Indica la ruta de la partición de datos
<b>ini_properties.data_partition_init_path</b>	String	Indica la ruta de la partición de datos inicial
<b>ini_properties.data_partition_size</b>	Integer	Tamaño de la partición de datos
<b>ini_properties.snap_storage_path</b>	String	Indica la ruta del snapshot a almacenar

## 2 EVENTOS DE CONFIGURACIÓN INICIAL DEL CONTEXTO

Atributo	Valores	Descripción
<b>Intall_app</b>	String	Instala la apk indicada (acepta el path)

<b>Timezone</b>	string	Configura la franja horaria del dispositivo
<b>power</b>		
<b>power.ac</b>	On off	Indica si el dispositivo está conectado a la corriente eléctrica
<b>Power.capacity</b>	integer	Indica la capacidad de carga de la batería
<b>Power.health</b>	Unknown Good Overheat Dead Overvoltage failure	Indica el estado de salud de la batería
<b>Power.present</b>	Boolean	Indica la presencia de batería o no en el dispositivo
<b>Power.status</b>	Unknown Charging Discharging not-charging full	Indica el estado de la batería

### 3 EVENTOS DE CONFIGURACIÓN DE CONTEXTO

Atributo	Valores	Descripción
<b>power</b>		
<b>power.ac</b>	On off	Indica si el dispositivo está conectado a la corriente eléctrica
<b>Power.capacity</b>	integer	Indica la capacidad de carga de la batería
<b>Power.health</b>	Unknown Good Overheat Dead Overvoltage failure	Indica el estado de salud de la batería
<b>Power.present</b>	Boolean	Indica la presencia de batería o no en el dispositivo
<b>Power.status</b>	Unknown Charging Discharging not-charging full	Indica el estado de la batería
<b>Gsm</b>		
<b>Gsm.Call</b>	String	Llamada al número indicado
<b>Gsm.Accept</b>	String	Acepta una llamada desde el número indicado
<b>Gsm.Busy</b>	String	Rechaza una llamada desde el número indicado

<b>Gsm.Cancel</b>	String	Cancela una llamada desde el número indicado
<b>Gsm.Data</b>	Unregistered Home Roaming Searching Denied Off on	Define el estado de la conexión de datos
<b>Gsm.Voice</b>	Unregistered Home Roaming Searching Denied Off on	Define el estado de la conexión de voz
<b>Sms</b>		
<b>Sms.number</b>	String	Indica el número al que se va a enviar el SMS
<b>Sms.message</b>	String	Contenido del SMS
<b>Geo</b>		
<b>geo.fix.longitude</b>	String	Indica la longitud en la geolocalización
<b>geo.fix.latitude</b>	String	Indica la latitud en la geolocalización
<b>geo.fix.altitude</b>	String	Indica la altitud en la geolocalización
<b>Geo.nmea</b>	String	Emulación NMEA
<b>Intents</b>		
<b>Intents.type</b>	String	Tipo de intent
<b>Intents.params</b>	String	Parámetros del intent
<b>Adb</b>		
<b>Adb.action</b>	String Enum = [ "connect_wifi", "lock_screen", "unlock_screen", "volumen_up", "go_home", "take_screenshot", "start_clock_app", "stop_clock_app", "start_wifi", "enable_wifi", "wifi_status", "enable_mobile_data", "disable_mobile_data" ]	Tipo de comando a ejecutar
<b>Monkey</b>		
<b>Monkey.path</b>	String	Path del script monkey

Monkey.package	String	Package sobre el que se ejecutará el script
----------------	--------	---

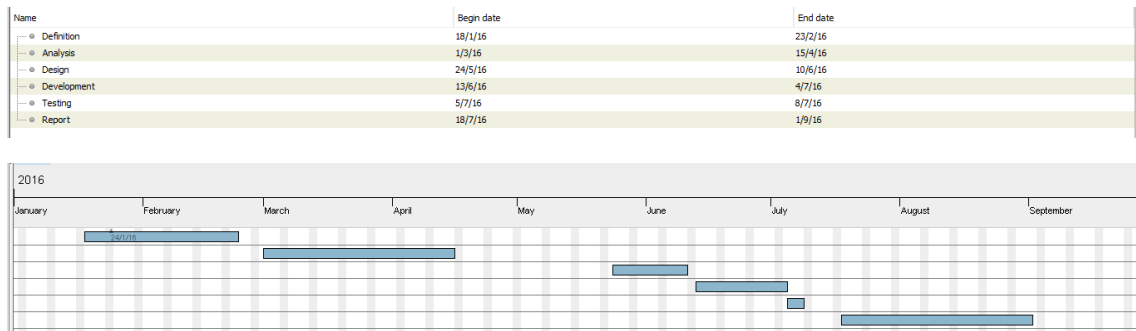
## Appendix C: Planning and budget

In order to comply with the regulations of MsC Thesis by Universidad Carlos III de Madrid, this appendix presents the Planning and Budget of the Thesis.

First, it is going to be presented a planning that defines this project in different tasks. It has been generated a typical Gantt chart presenting in a graphical form the duration of each phase.

The resume of the phases is shown in Figure X with its start and end date. Gantt chart with detailed information is shown in Figure X+1.

	Duration (days)	Start Date	End Date
Definition of goals	27	16/01/2016	23/02/2016
Analysis	34	01/03/2016	15/04/2016
Design	14	24/05/2016	10/06/2016
Development	16	13/06/2016	04/07/2016
Testing	4	5/07/2016	10/07/2016
Report	34	18/07/2016	1/09/2016



Last, it is going to be presented the budget. It is broken down into various estimations by type cost:

- Personal cost: 31.600 €
- Hardware cost: 30 €
- Software cost: 0 €
- Indirect cost: 301 €
- Total cost: 31.600 €

Note: VATs are included.

Concept	Time (hours)	Fees	RRHH cost
Security Engineer	500	50 €/hour	25.000 €
PhD	120	55 €/hour	6.600 €
TOTAL			31.600 €



Concept	Units	Unit price	Estimated life time	Airtime (months)	Cost
Acer Aspire 5740	1	600 €	60 months	4,3	30 €
TOTAL					30 €

Concept	Units	Unit price	Estimated life time	Airtime	Cost
Ubuntu Linux	1	0 €	-	4,3	0 €
Sublime Text 3	1	0 €	-	4,3	0 €
SmartGit Hg 7.1	1	0 €	-	4,3	0 €
Github	1	0 €	-	4,3	0 €
Bitbucker	1	0 €	-	4,3	0 €
MiKTeX	1	0 €	-	4,3	0 €
TeXnic Center	1	0 €	-	4,3	0 €
Adobe Reader	1	0 €	-	4,3	0 €
Google Chrome	1	0 €	-	4,3	0 €
TOTAL					0 €

Concept	Monthly price	Airtime (months)	Cost
Electricity	40 €	4,3	172 €
Internet	30 €	4,3	129 €
TOTAL			301 €

Concept	Amount
Personal cost	31.600 €
Hardware cost	30 €
Software cost	0 €
Indirect cost	301 €
TOTAL	31.931 €