UNIVERSIDAD INTERNACIONAL DE LA RIOJA

Universidad Internacional de La Rioja
Máster universitario en Seguridad Informática

Malware en Android y medidas de prevención.

Trabajo Fin de Máster

presentado por: Villanova Pascual, Oscar

Director/a: Díaz Vico, Jesús

Ciudad: Tarragona
Fecha: 27/01/2016

Resumen

En este trabajo se hace un estudio en profundidad sobre la situación actual de la plataforma Android focalizándolo en la seguridad. A continuación se analiza, de forma general, el malware en Android y posteriormente las herramientas de prevención disponibles en el mercado para la citada plataforma.

Seguidamente, nos centramos en el objetivo principal del trabajo. Hacer una evaluación sobre las herramientas de prevención existentes en el mercado y determinar si son eficaces contra el malware, además de comprobar cómo afectan al rendimiento del dispositivo. En esta parte del trabajo se hace una selección del malware a testear, intentando que éste sea representativo de la gran variedad que hay. Para continuar, se hace lo mismo con las herramientas de prevención y, seguidamente se hacen los test comprobando la efectividad de las herramientas y mostrando los resultados obtenidos de las pruebas realizadas. El trabajo concluye con una guía de buenas prácticas de seguridad, conclusiones y trabajo futuro.

Palabras Clave: Android, malware, protección, prevención, evaluación.

Abstract

This project makes a deep study about the current situation of the Android platform, focusing on security. After, Android malware is analysed in general terms and, subsequently the set of prevention tools available in the market for this platform.

Later on, we focus on the main goal of this project. To make an evaluation of the commercially available prevention tools, and determine whether they are effective against malware, as well as testing how they affect the device performance. In this part of the project the malware to be tested is selected, trying to make it representative of the great variety available. Following, the selection is done with the prevention tools, testing them to check its effectiveness, and showing the results obtained. The study ends with a best practices manual, conclusions and future work.

Keywords: Android, malware, protection, prevention, evaluation.

ESTRUCTURA CAPÍTULOS

1. INTRODUCCIÓN 1

2. SITUACIÓN ACTUAL 2

2.1. ANDROID 3

2.1.1. HISTORIA 3

2.1.2. VERSIONES 4

2.2. ARQUITECTURA DE ANDROID 5

2.3. MODELO DE SEGURIDAD DE ANDROID 10

2.3.1. AISLAMIENTO DE APLICACIONES (SANDBOX) 11

2.3.2. PERMISOS 12

2.3.3. PROCEDENCIA DE APLICACIONES 15

2.3.4. VERIFICACIÓN DE APLICACIONES 16

2.3.5. POLÍTICA PARA DESARROLLADORES DE PLAY STORE 17

2.3.6. ELIMINACIÓN REMOTA DE APLICACIONES 18

2.3.7. CIFRADO DE DATOS 18

2.3.8. COMUNICACIONES SEGURAS 19

2.3.9. CONTROL DE ACCESO FÍSICO 20

2.4. APLICACIONES EN ANDROID 21

2.5. EL MALWARE EN ANDROID 24

2.5.1. CONCEPTOS Y DEFINICIONES 25

2.5.2. FINES DEL MALWARE 29

2.5.3. CASOS DE MALWARE EN ANDROID. 29

2.6. FORMAS DE INFECCIÓN 34

2.7. HERRAMIENTAS DE PREVENCIÓN 35

2.7.1. ¿ES NECESARIO DISPONER DE ALGUNA HERRAMIENTA DE PREVENCIÓN? 35

2.7.2. CARACTERÍSTICAS GENERALES DE HERRAMIENTAS DE PREVENCIÓN 38

2.7.3. LISTADO DE HERRAMIENTAS 40

3. OBJETIVO Y METODOLOGÍAS DE EVALUACIÓN 43
3.1. OBJETIVO 43
3.2. METODOLOGÍA EFECTIVIDAD HERRAMIENTAS Y RENDIMIENTO 44
3.3. METODOLOGÍA ANÁLISIS CONSUMO DE BATERÍA 44

4. MALWARE A ESTUDIAR 45
4.1. TIPOLOGÍAS DE MALWARE MÁS COMÚN 45
4.2. SELECCIÓN DE MALWARE 47

5. SELECCIÓN HERRAMIENTAS DE PREVENCIÓN A EVALUAR 51

6. RESULTADOS EVALUACIÓN 52
6.1. EFECTIVIDAD HERRAMIENTAS 52
6.2. IMPACTO EN EL RENDIMIENTO DEL SISTEMA 55
6.2.1. CONSUMO DE CPU Y MEMORIA RAM 55
6.2.2. CONSUMO DE BATERIA 71
6.2.3. RESUMEN EVALUACIÓN RENDIMIENTO Y CONSUMO DE BATERÍA 77

7. GUÍA DE BUENAS PRÁCTICAS 79

8. CONCLUSIONES Y TRABAJO FUTURO 80

9. REFERENCIAS BIBLIOGRÁFICAS Y ENLACES 83
9.1. ORÍGENES DE FIGURAS 87

10. ANEXOS 87

INDICE DE FIGURAS

Figura 2.1 – Fragmentación de Android.
Figura 2.2 – Capas arquitectura Android.
Figura 2.3 – Sandbox en Android.
Figura 2.4 – Configuración Verificación de aplicaciones en Android 4.4.
Figura 2.5 – Pantallas de activación cifrado dispositivo.
Figura 2.6 – Comparativa de la estructura interna de un archivo .class y un archivo .dex
Figura 2.7 – Dispositivos por Sistema Operativo en el mundo.
Figura 2.8 – Malware detectado en 2014 según S.O.
Figura 2.9 – Casos de malware en Android, por países.
Figura 2.10 – Casos de malware en Android, por tipos.
Figura 2.11 – Casos de malware en Android.
Figura 2.12 – Casos de malware en Android por tipos.
Figura 4.1 – Ranking familias Malware año 2012.
Figura 5.1 – Logos de las empresas de herramientas de prevención de evaluar.
Figura 6.1 – Gráfica resultados porcentaje detección de malware.
Figura 6.2 – Consumo de CPU y memoria sin herramienta de prevención instalada.
Figura 6.3 – Consumo de CPU y memoria herramienta de Avast.
Figura 6.4 – Detalle del proceso Avast Mobile Security.
Figura 6.5 – Consumo de CPU y memoria herramienta de de AVG.
Figura 6.6 – Detalle del proceso AVG.
Figura 6.7 – Consumo de CPU y memoria herramienta de Avira.
Figura 6.8 – Detalle del proceso Avira.
Figura 6.9 – Consumo de CPU y memoria herramienta de Bitdefender.
Figura 6.10 – Detalle del proceso Bitdefender.
Figura 6.11 – Consumo de CPU y memoria herramienta de ESET.
Figura 6.12 – Detalle del proceso ESET.
Figura 6.13 – Consumo de CPU y memoria herramienta de IKARUS.
Figura 6.14 – Detalle del proceso IKARUS.
Figura 6.15 – Consumo de CPU y memoria herramienta de Intel Security.
Figura 6.16 – Detalle del proceso Intel Security.
Figura 6.17 – Consumo de CPU y memoria herramienta de Kaspersky.
Figura 6.18 – Detalle del proceso Kaspersky.
Figura 6.19 – Consumo de CPU y memoria herramienta de SOPHOS.
Figura 6.20 – Detalle del proceso SOPHOS.

Figura 6.21 – Consumo de CPU y memoria herramienta de Trend Micro.
Figura 6.22 – Detalle del proceso Trend Micro.
Figura 6.23 – Consumo de batería del dispositivo ejecutando la herramienta de AVAST.
Figura 6.24 – Consumo de batería ejecutando la herramienta de AVG.
Figura 6.25 – Consumo de batería ejecutando la herramienta de AVIRA.
Figura 6.26 – Consumo de batería ejecutando la herramienta de BitDefender.
Figura 6.27 – Consumo de batería ejecutando la herramienta de ESET.
Figura 6.28 – Consumo de batería ejecutando la herramienta de IKARUS.
Figura 6.29 – Consumo de batería ejecutando la herramienta de Intel Security.
Figuras 6.30 – Consumo de batería ejecutando la herramienta de KASPERSKY.
Figuras 6.31 – Consumo de batería ejecutando la herramienta de SOPHOS.
Figura 6.32 – Consumo de batería ejecutando la herramienta de Trend Micro.

INDICE DE TABLAS

Tabla 2.1 – Versiones de Android
Tabla 2.2 – Posibles permisos a otorgar a una aplicación.
Tabla 2.3 – Permisos simplificados en Android
Tabla 2.4 – Herramientas de protección para Android
Tabla 4.1 – Malware seleccionado para el testeo de las herramientas de prevención
Tabla 6.1 – Tabla resumen evaluación del rendimiento y consumo de batería

1. INTRODUCCIÓN

Este Trabajo Fin de Master (TFM) está justificado por la necesidad de hacer una recopilación enfocada a la temática de seguridad de la plataforma Android, centrándose en el malware en Android y medidas de prevención contra el malware.

Teniendo en cuenta que el primer dispositivo con Android salió al mercado el 22 de octubre de 2008 y la penetración en el mundo de los dispositivos móviles ha sido realmente extraordinaria (en agosto de 2015 el ecosistema Android contaba con 24.093 dispositivos diferentes y había más de 1.400 millones de usuarios con Android), esto ha conllevado que, para los ciberdelincuentes, este entorno sea un campo abonado para sus intereses. De esta forma, en este TFM se pretende, dar visibilidad a las medidas de seguridad para la plataforma establecidas hasta la fecha. A su vez, se explica de forma pormenorizada el malware que existe para Android, partiendo de los conceptos y definiciones del propio malware, siguiendo con una explicación de los fines que el malware persigue, continuando por los casos que han aparecido, basándonos en informes de Google (empresa propietaria de Android) y de empresas que desarrollan herramientas de prevención y, finalizando en las diferentes formas de infección.

Ante esta perspectiva de crecimiento, al igual que los ciberdelincuentes, ha habido muchas empresas de seguridad informática que han visto que la plataforma Android, sería un buen espacio donde desarrollar sus negocios de seguridad y crearon, a la vez que Android se desarrollaba, herramientas de prevención contra el malware (la primera salió al mercado en noviembre de 2008). Sobre este tema existe una fuerte discusión de si realmente son (o no) necesarias este tipo de herramientas y hay opiniones en ambos sentidos. Por nuestra parte consideramos que sí, que son necesarias ya que el malware existe y no siempre se siguen las mejores prácticas en cuanto a la seguridad. Es más, hay muchos usuarios de Android que ni tan siquiera son conscientes que pueden ser objeto de ataques.

Planteado el escenario, pasamos a detallar el objetivo principal del TFM, que consiste en, evaluar un conjunto de herramientas de prevención existentes en el mercado y determinar si son eficaces contra el malware. Además, se comprobará cómo afectan al rendimiento del dispositivo, en cuanto a consumo de CPU, memoria y durabilidad de la batería.

La metodología a utilizar para realizar la evaluación, se basa en realizar las pruebas en un entorno virtual, transfiriendo las muestras de malware (previamente seleccionadas para que representen un amplio espectro de funcionalidades) a ese entorno para comprobar si las herramientas de prevención instaladas son realmente efectivas. Además, se habrán instalado herramientas de monitorización del sistema para comprobar cómo afectan las citadas medidas de prevención al rendimiento del sistema (en cuanto a consumo de CPU y memoria) y a la durabilidad de la batería.

Con todo lo expuesto en el TFM, se creará una guía de buenas prácticas, de tal forma que el usuario de Android esté prevenido y, en la medida de lo posible, protegido contra el malware. Se pondrá de relieve que el centro de atención para estar seguro son los comportamientos de las personas y el sentido común, ya que no sólo nos podemos basar en la tecnología para defendernos de los ciberdelincuentes. El TFM se terminará con las conclusiones extraídas del mismo, sobretodo basándonos en la parte experimental y reflexionando sobre trabajos futuros en relación a la temática planteada.

2. SITUACIÓN ACTUAL

En este capítulo vamos a realizar un estudio pormenorizado de todo lo relativo a la plataforma Android, empezando por definirlo y explicar cuándo, cómo y dónde aparece y, continuando la exposición con las diferentes versiones aparecidas hasta el momento. Seguidamente detallaremos la arquitectura de la plataforma y nos centraremos en el modelo de seguridad, ya que las medidas que Google propone para la plataforma son amplias y variadas.

Debido al éxito que ha tenido Android, se ha generado multitud de malware de diferente índole para esta plataforma y, en este ámbito, pasaremos a aclarar conceptos y definiciones del malware, además, ilustraremos con qué fines se produce, las formas de infección posibles y mostraremos información ofrecida tanto por Google como por Symantec donde se muestra la divergencia de sus análisis en cuanto a casos de malware que se dan en Android. Además, con la aparición del malware se ha creado una industria de herramientas de prevención contra el mismo. En este escenario, y en primer lugar, nos plantearemos si es necesario disponer de alguna herramienta adicional para protegernos del malware (teniendo en cuenta que Google hace un despliegue amplio de medidas de seguridad) y, a continuación explicaremos

las características generales de las herramientas disponibles en el mercado y finalizaremos el estudio del capítulo con un listado de herramientas tanto gratuitas como de pago.

2.1. ANDROID

Android es una pila de software pensada inicialmente para teléfonos móviles (smartphones) que incluye un sistema operativo, middleware y una capa aplicaciones para que el teléfono pueda realizar funciones más allá de las de llamar y enviar/recibir mensajes como han sido históricamente las funciones base de un teléfono.

2.1.1. HISTORIA

Mirando la historia de esta plataforma, tenemos que Android era un sistema operativo para móviles prácticamente desconocido propiedad de una empresa llamada Android Inc. hasta que en 2005 Google lo compró. Andy Rubin, el creador de Android, pasó a trabajar como director de la división de móvil de Google hasta Octubre de 2014[1].

El 5 de noviembre de 2007[2], Google anunció la creación de la Open Handset Alliance, un consorcio de 47 empresas de hardware, software y telecomunicaciones dedicadas al fomento de estándares abiertos para dispositivos móviles, entre éstas estaban, entre otras, HTC, Samsung, T-Mobile, Intel, Texas Instruments, China Mobile, etc. A la vez, el día 12 de noviembre[3] se proporcionó la primera versión de Android, junto con el Android Software Development Kit (SDK) para que los programadores empezaran a crear sus aplicaciones para este sistema. La idea era que dichas empresas fabricarían teléfonos Android y promoverían una plataforma móvil de código libre.

La alianza ha rendido sus frutos a nivel comercial. Google provee el software (en el que su buscador, su correo, sus mapas y videos son la parte central) y los fabricantes compiten entre ellos tratando de diseñar la mejor plataforma para ejecutar el código Android. Gracias a ello, por ejemplo, Samsung es líder en el mercado mundial de teléfonos inteligentes[4].

El primer móvil con el sistema operativo Android fue el HTC Dream y se puso a la venta en octubre de 2008. A partir de ese momento, tanto Android como los dispositivos que se han desarrollado alrededor de este ecosistema han sido

numerosos y con mucho éxito; ya no sólo hablando de teléfonos inteligentes, sino ampliando el abanico de productos con otros dispositivos como pueden ser tabletas, ordenadores portátiles, netbooks, relojes de pulsera, Google TV, auriculares, etc.

2.1.2. VERSIONES

En cuanto a las versiones de Android también han aparecido unas cuantas desde la primera - como algo curioso, las versiones de Android reciben, en inglés, el nombre de diferentes postres o dulces-. En cada versión, el postre o dulce elegido empieza por una letra distinta, conforme a un orden alfabético. La Tabla 2.1 presenta todas las versiones que han aparecido hasta la fecha[5].

VERSIÓN	NOMBRE	FECHA DE LANZAMIENTO
1.0	Apple Pie (Tarta de manzana)	23/09/2008
1.1	Banana Bread (Pan de plátano)	9/02/2009
1.5	Cupcake (Panque)	27/04/2009
1.6	Donut (Rosquilla)	15/09/2009
2.0 - 2.1	Éclair (Pepito)	26/10/2009 – 12/01/2010
2.2	Froyo (Yogur helado)	20/05/2010
2.3	Gingerbread (Pan de jengibre)	6/12/2010
3.0 - 3.1 - 3.2	Honeycomb (Panal de miel)	22/02/2011 – 10/05/2011 – 15/07/2011
4.0	Ice Cream Sandwich (Sándwich de helado)	18/10/2011
4.1 - 4.2 - 4.3	Jelly Bean (Gominola)	9/07/2012 – 13/11/2012 – 24/07/2013
4.4	KitKat (Kit Kat)	31/10/2013
5.0/5.1	Lollipop (Piruleta)	12/11/2014 – 9/03/2015
6.0	Marshmallow (Malvavisco)	05/10/2015

Tabla 2.1 – Versiones de Android

Debido a la multitud de fabricantes y dispositivos de Android (hemos de ser conscientes que según un reciente estudio, en Agosto de 2015 el ecosistema Android contaba con 24.093 dispositivos diferentes fabricados por 1.294 empresas distintas[6] y, que en el mundo hay más de 1.400 millones de usuarios con Android[7]) aparece el problema de la fragmentación, es decir, que aunque Google continúa evolucionando Android, estas mejoras (a nosotros nos interesan sobre todo las relacionadas con la

seguridad) no llegan a todos los dispositivos desplegados, debido en parte a limitaciones de hardware, por dejadez y/o falta de prioridad por parte del fabricante. Además, la fragmentación provoca que los desarrolladores tengan más complicada su tarea para que el software llegue al mayor número de dispositivos teniendo en cuenta que cada versión de Android tiene su entorno de desarrollo que evoluciona y el hardware, ya sea procesador, memoria, pantalla, etc. es diferente en cada dispositivo.

Ante esta problemática, Google publica una página web[8], destinada a los desarrolladores donde informa sobre el número relativo de dispositivos que comparten unas determinadas características como la versión de Android que ejecutan o tamaño de las pantallas. Esta información, dice Google, les puede ayudar a priorizar los esfuerzos para que sus aplicaciones se puedan ejecutar en el mayor número de dispositivos. La figura 2.1 consta de una tabla y un gráfico sobre la fragmentación en cuanto a las versiones de Android a fecha de 2 de noviembre de 2015.

Version	Codename	API	Distribution
2.2	Froyo	8	0.2%
2.3.3 - 2.3.7	Gingerbread	10	3.8%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	3.3%
4.1.x	Jelly Bean	16	11.0%
4.2.x		17	13.9%
4.3		18	4.1%
4.4	KitKat	19	37.8%
5.0	Lollipop	21	15.5%
5.1		22	10.1%
6.0	Marshmallow	23	0.3%

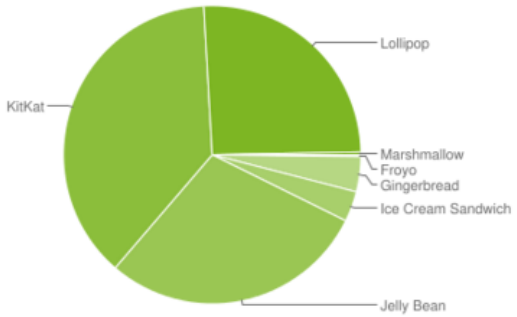


Figura 2.1 – Fragmentación de Android. Extraída de [a]

2.2. ARQUITECTURA DE ANDROID

Como ya se ha mencionado anteriormente, Android es una plataforma para dispositivos móviles que contiene una pila de software donde se incluye un sistema operativo, middleware y aplicaciones básicas para el usuario. A continuación se muestra la Figura 2.2 donde se diferencian las diferentes capas que componen la

plataforma. Hay que tener en cuenta que cada capa utiliza servicios ofrecidos por las anteriores, y ofrece, a su vez, sus propios servicios a las capas de niveles superiores.

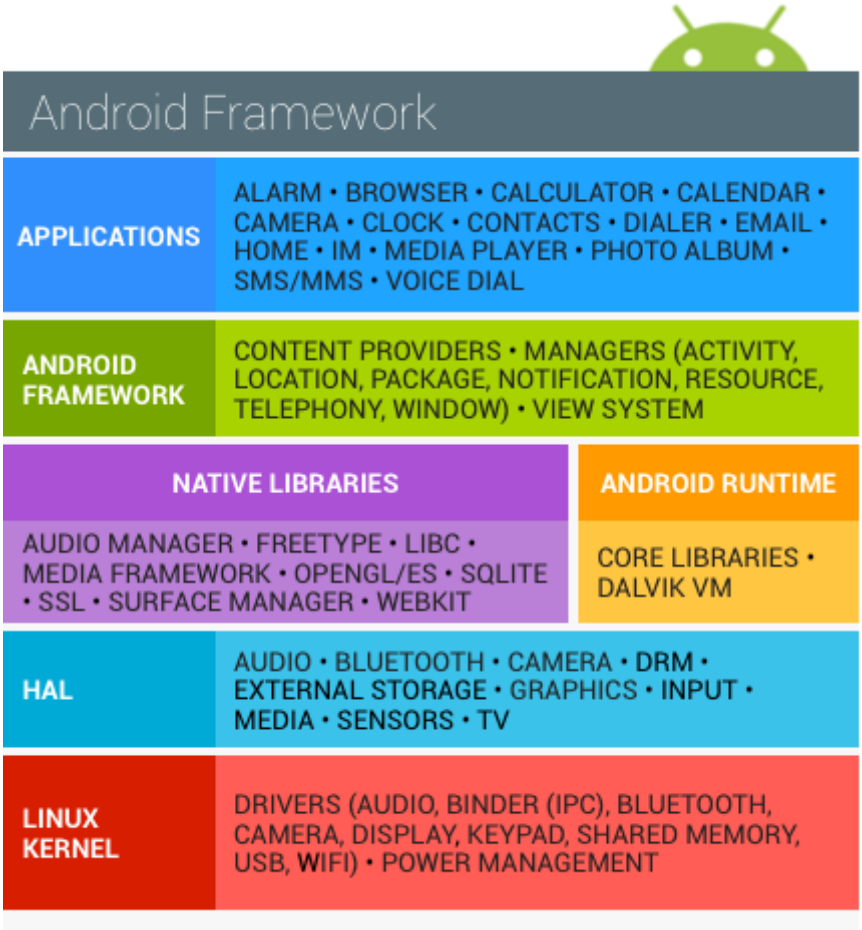


Figura 2.2 – Capas arquitectura Android. Extraída de [b]

Seguidamente vamos a describir cada una de las capas que componen la arquitectura[9][10][11].

Núcleo (kernel) de Linux: Android está construido sobre el **núcleo de Linux**, pero se ha modificado para adaptarse a dispositivos móviles. Esta elección está basada en la excelente potabilidad, flexibilidad y seguridad que Linux presenta. Recordemos que el Kernel de Linux está bajo la licencia GPL[12], así que en consecuencia Android también.

El núcleo se encarga de gestionar los diferentes recursos del teléfono como pueden ser energía, memoria, etc. y del sistema operativo en sí: procesos, elementos de comunicación, etc.

Hardware Abstraction Layer (HAL): Este componente es aquel que permite la independencia del hardware. Quiere decir que Android está construido para ejecutarse en cualquier dispositivo móvil sin importar su arquitectura física. El HAL actúa como una **arquitectura genérica** que representa a todos los posibles tipos de hardware existentes en el mercado. Aunque por el momento no hay estándares de construcción en el hardware de dispositivos móviles, el HAL permite que cada fabricante ajuste sus preferencias para que Android sea funcional sobre su tecnología.

Librerías nativas: La siguiente capa se corresponde con las librerías utilizadas por Android. Éstas han sido escritas utilizando C/C++ y proporcionan a Android la mayor parte de sus capacidades más características. Hay que tener en cuenta que normalmente están hechas por el fabricante del dispositivo y están compiladas para la arquitectura hardware específica del teléfono. Junto al núcleo (kernel) basado en Linux, estas librerías constituyen el corazón de Android.

El objetivo de las librerías es proporcionar funcionalidad a las aplicaciones para tareas que se repiten con frecuencia, evitando tener que codificarlas cada vez y garantizando que se llevan a cabo de la forma más eficiente. Entre las librerías más importantes ubicadas aquí, se pueden encontrar las siguientes:

- ***Librería libc**:* Incluye todas las cabeceras y funciones según el estándar del lenguaje C. Todas las demás librerías se definen en este lenguaje.
- ***Librería Surface Manager**:* Es la encargada de componer los diferentes elementos de navegación de pantalla. Gestiona también las ventanas pertenecientes a las distintas aplicaciones activas en cada momento.
- ***OpenGL/SL y SGL:*** Representan las librerías gráficas y, por tanto, sustentan la capacidad gráfica de Android. OpenGL/SL maneja gráficos en 3D y permite utilizar, en caso de que esté disponible en el propio dispositivo móvil, el hardware encargado de proporcionar gráficos 3D. Por otro lado, SGL proporciona gráficos en 2D, por lo que será la librería más habitualmente utilizada por la mayoría de las aplicaciones. Una característica importante de la capacidad gráfica de Android es que es posible desarrollar aplicaciones que combinen gráficos en 3D y 2D.
- ***Librería Media Libraries:*** Proporciona todos los códecs necesarios para el contenido multimedia soportado en Android (vídeo, audio, imágenes estáticas y animadas, etc.)
- ***FreeType:*** Permite trabajar de forma rápida y sencilla con distintos tipos de fuentes de texto.

- ***Librería SSL:*** Posibilita la utilización de dicho protocolo para establecer comunicaciones seguras.
- ***Librería SQLite:*** Creación y gestión de bases de datos relacionales.
- ***Librería WebKit:*** Proporciona un motor para las aplicaciones de tipo navegador y forma el núcleo del actual navegador incluido por defecto en la plataforma Android.

Entorno de ejecución: El entorno de ejecución de Android no se considera una capa en sí mismo, dado que también está formado por librerías. Aquí encontramos las librerías con las funcionalidades habituales de Java así como otras específicas de Android.

El componente principal del entorno de ejecución de Android es la máquina virtual Dalvik. Las aplicaciones se codifican en Java y son compiladas en un formato específico para que esta máquina virtual las ejecute. La ventaja de esto es que las aplicaciones se compilan una única vez y de esta forma estarán listas para distribuirse con la total garantía de que podrán ejecutarse en cualquier dispositivo Android que disponga de la versión mínima del sistema operativo que requiera la aplicación.

Cabe aclarar que Dalvik es una variación de la máquina virtual de Java, por lo que no es compatible con el bytecode Java. Java se usa únicamente como lenguaje de programación, y los ejecutables que se generan con el SDK de Android tienen la extensión .dex que es específico para Dalvik, y por ello no podemos correr aplicaciones Java en Android ni viceversa.

Dalvik está optimizada para requerir poca memoria y está diseñada para que permita ejecutar varias instancias de la máquina virtual simultáneamente, delegando el control y la gestión de memoria al sistema operativo subyacente. Se debe resaltar, que a diferencia de una máquina virtual Java, la cual se basa en registros, Dalvik está completamente basada en una estructura de pilas.

Framework de Aplicaciones: Representa fundamentalmente el conjunto de herramientas de desarrollo de cualquier aplicación, es decir, está formada por todas las clases y servicios que utilizan directamente las aplicaciones para realizar sus funciones. La mayoría de los componentes de esta capa son librerías Java que acceden a los recursos de las capas inferiores a través de la máquina virtual Dalvik.

Entre las API más importantes ubicadas aquí, se pueden encontrar las siguientes:

- ***Activity Manager:*** Conjunto de API que gestiona la pila de actividades y el ciclo de vida de las aplicaciones en Android.
- ***Windows Manager:*** Gestiona las ventanas de las aplicaciones. Básicamente crea las superficies de pantalla (utiliza la librería Surface Manager) que posteriormente pasarán a ser ocupadas por las actividades.
- ***Telephony Manager:*** Incluye todas las API vinculadas a las funcionalidades propias del teléfono, realizar llamadas o enviar y recibir SMS/MMS.
- ***Content Provider:*** Permite a cualquier aplicación compartir sus datos con las demás aplicaciones de Android. Por ejemplo, gracias a esta API la información de contactos, agenda, mensajes, etc. será accesible para otras aplicaciones.
- ***View System:*** Proporciona un gran número de elementos para poder construir interfaces de usuario (GUI), como listas, mosaicos, botones, "check-boxes", tamaño de ventanas, control de las interfaces mediante teclado, etc. Incluye también algunas vistas estándar para las funcionalidades más frecuentes.
- ***Location Manager:*** Posibilita a las aplicaciones la obtención de información de localización y posicionamiento ya sea mediante redes GPS o redes disponibles, así como trabajar con mapas.
- ***Notification Manager:*** Mediante el cual las aplicaciones, usando un mismo formato (mostrar alertas en la barra de estado), comunican al usuario eventos que ocurran durante su ejecución: una llamada entrante, un mensaje recibido, conexión Wi-Fi disponible, ubicación en un punto determinado, etc. Si llevan asociada alguna acción, en Android denominada **Intent**, (por ejemplo, atender una llamada recibida) ésta se activa mediante un simple clic. Un dato importante de esta biblioteca es que también permite configurar sonidos, activar el vibrador o utilizar los LEDs del teléfono en caso de tenerlos.
- ***Package Manager:*** Esta biblioteca permite obtener información sobre las aplicaciones instaladas en el dispositivo Android, además de gestionar la instalación de nuevas aplicaciones.
- ***Resource Manager:*** Con esta librería podremos gestionar todos los elementos que forman parte de la aplicación y que están fuera del código, es decir, cadenas de texto traducidas a diferentes idiomas, imágenes, sonidos o diseños.
- ***Sensor Manager***: Nos permite manipular los elementos de hardware del teléfono como el acelerómetro, giroscopio, sensor de luminosidad, sensor de campo magnético, brújula, sensor de presión, sensor de proximidad, sensor de temperatura, etc.
- ***Cámara***: Con esta librería podemos hacer uso de la/s cámara/s del dispositivo

para tomar fotografías o para grabar vídeo.

- ***Multimedia***: Permiten reproducir y visualizar audio, vídeo e imágenes en el dispositivo.
- ***XMPP Service:*** Colección de API para utilizar este protocolo de intercambio de mensajes basado en XML.

Aplicaciones: Este nivel contiene, tanto las aplicaciones incluidas (preinstaladas) por defecto de Android como aquellas que el usuario vaya instalando posteriormente, ya sean de terceras empresas o de su propio desarrollo. Todas estas aplicaciones utilizan los servicios, las API y librerías de los niveles inferiores.

En esta capa también encontraremos la aplicación principal del sistema: Inicio (Home) o lanzador (launcher), y es la que permite ejecutar otras aplicaciones mediante una lista y mostrando diferentes escritorios donde se pueden colocar accesos directos a aplicaciones o incluso widgets, que son también aplicaciones de esta capa.

2.3. MODELO DE SEGURIDAD DE ANDROID

La seguridad en Android abarca desde el despliegue hasta la ejecución de la aplicación. La mayoría de las medidas de seguridad entre el sistema y las aplicaciones derivan de los estándares de Linux, cuyo kernel, constituye el núcleo principal de Android. Básicamente se aplica el principio de mínimo privilegio por defecto, donde cada aplicación (o app) únicamente dispone acceso a sus propios componentes. Las apps por tanto, deben definir y solicitar permiso para acceder a los recursos y datos compartidos en los que están interesadas.

El sistema notifica al usuario los permisos que requiere en el momento de la instalación de la app, en ningún caso se solicitará autorización al usuario durante la ejecución de la aplicación (las excepciones son: el acceso al USB y a partir de la versión 6.0, recientemente publicada). Hasta ahora, si se instalaba una app, ésta dispondrá de acceso a todos los permisos solicitados, en caso que el usuario no esté interesado en proporcionar ciertos permisos a una app, la única opción disponible es no instalar dicha app. Esto cambia a partir de la versión 6.0, donde también es posible, eliminar o dar permisos a una app una vez instalada. Por ejemplo, un usuario puede denegar el acceso a una app a utilizar su ubicación si no lo considera necesario[13].

2.3.1. AISLAMIENTO DE APLICACIONES (SANDBOX)

Cada vez que se instala una aplicación en Android se crea un usuario Linux para ésta, de forma que este usuario y, por tanto, la aplicación asociada, sólo tiene acceso a sus propios recursos. De esta manera, cada proceso en Android, que tiene asignada su propia máquina virtual y se ejecuta de forma aislada al resto de apps, - en lo que se conoce como *sandbox*[14] - proporciona un entorno seguro de ejecución y, por defecto no tiene permiso para realizar ninguna operación o comportamiento que pueda impactar negativamente en la ejecución de otras aplicaciones o del sistema mismo. Por ejemplo, acciones como leer o escribir ficheros privados de otro usuario no están permitidas. La única forma de poder saltar estas restricciones impuestas por Android, es mediante la declaración explícita de un permiso que autorice a llevar a cabo una determinada acción por defecto prohibida.

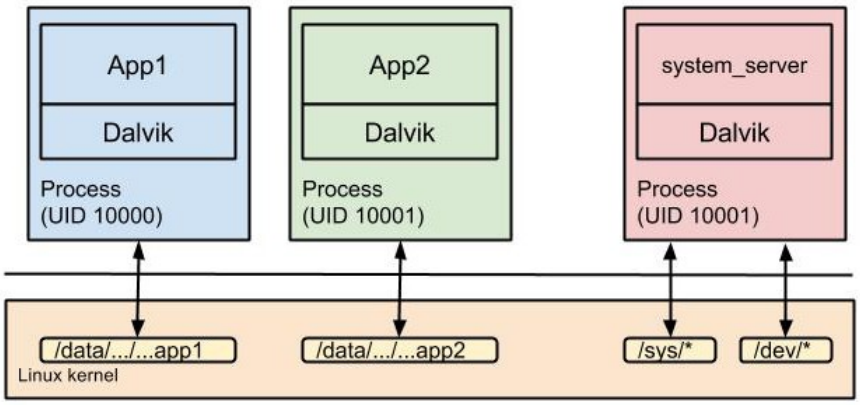


Figura 2.3 – Sandbox en Android. Extraída de [c]

Como ya hemos mencionado, el modelo de seguridad de Android se basa parcialmente en la utilización de contenedores (o sandboxes) para cada aplicación móvil (o app). La implementación previa a la versión 4.3 de Android empleaba un identificador de usuario (UID) único de Linux, generado en el momento de la instalación de la app, para restringir el entorno de ejecución de cada app. Sin embargo, a partir de la versión 4.3 de Android, se emplea también SELinux (SecurityEnhanced Linux)[15] para restringir los límites y capacidades asociadas a una app y a su contenedor.

SELinux proporciona un modelo de seguridad conocido como MAC (Mandatory Access

Control), donde el sistema especifica qué sujetos pueden tener acceso a objetos de datos específicos, frente al modelo de seguridad tradicional DAC (Discretionary Access Control), dónde es el propietario de los objetos el que define que sujetos pueden acceder al objeto, asimismo SELinux proporciona la definición asociada de políticas de seguridad[16].

2.3.2. PERMISOS

Con respecto a la ejecución, como ya hemos mencionado antes, cada aplicación Android se ejecuta dentro de un proceso separado, además, cada uno de estos procesos tiene un ID de usuario único y permanente (asignado en el momento de la instalación). Si una aplicación desea acceder a recursos o datos compartidos debe utilizar los permisos declarados, en caso contrario, si una aplicación intenta hacer uso de un permiso no declarado se creará una excepción de permiso y la aplicación se detendrá. A continuación en la Tabla 2.2 presentamos una lista (no exhaustiva) de permisos que se pueden otorgar a una aplicación[17]. Hay que tener en cuenta que la sintaxis correcta es “android.permission.Nombre_permiso” y que todos ellos están definidos en la clase Manifest.permission.

NOMBRE PERMISO	DESCRIPCIÓN
ACCESS_FINE_LOCATION	Permite a una aplicación acceder a la localización exacta del terminar
ACCESS_WIFI_STATE	Permite a una aplicación acceder a información sobre las redes Wifi
BATTERY_STATS	Permite a una aplicación a recopilar estadísticas de la batería.
BLUETOOTH	Permite a las aplicaciones conectar a los dispositivos asociados a través de Bluetooth
CALL_PHONE	Permite a una aplicación iniciar una llamada sin pasar por el marcador para que el usuario pueda confirmar
CAMERA	Da acceso a la cámara del dispositivo

CHANGE_NETWORK_STATE	Permite cambiar el estado de conectividad de red
INSTALL_PACKAGES	Permiso para instalar aplicaciones
INTERNET	Permiso para abrir sockets de red
NFC	Permiso para realizar operaciones de E/S sobre NFC
READ_LOGS	Permisos para leer los registros de llamadas
READ_SMS	Permisos para leer SMS
REBOOT	Requerido para poder reiniciar el dispositivo
SEND_SMS	Permisos para enviar SMS
VIBRATE	Da acceso al vibrador del dispositivo
WRITE_CALENDAR	Permite a la aplicación escribir en el calendario del usuario
WRITE_SETTINGS	Permiso para leer o escribir la configuración del sistema.

Tabla 2.2 – Posibles permisos a otorgar a una aplicación.

Como vemos la cantidad de permisos disponibles, y los componentes o recursos que abarcan, es amplio y variado (en total hay 136 permisos diferentes)[17]. Por este motivo y para intentar simplificar la gestión de los mismos –tanto para los desarrolladores, como para la comprensión por parte de los usuarios - , Google creó en junio de 2014, los permisos simplificados (o grupos de permisos de Android). Esto implica que en lugar de mostrarse (en el momento de la instalación) los permisos individuales solicitados por cada app, se han agrupado en función de las capacidades o funciones específicas a las que está asociado cada uno de los permisos, añadiéndose un icono descriptivo asociado[16]. Esta agrupación ha conseguido agrupar todos los permisos en nueve categorías o grupos de permisos. La Tabla 2.3 muestra los grupos que hay actualmente[18]. Hay que tener en cuenta que la sintaxis correcta es "android.permission-group.Nombre_grupo_permisos" y que todos ellos están definidos

en la clase Manifest.permission_group.

NOMBRE GRUPO PERMISOS	DESCRIPCIÓN
CALENDAR	Se utiliza para los permisos de ejecución relacionados con el calendario del usuario.
CAMERA	Se utiliza para los permisos que están asociados con el acceso a la cámara o la captura de imágenes / vídeo desde el dispositivo.
CONTACTS	Se utiliza para los permisos de ejecución relacionados con los contactos del usuario y perfil.
LOCATION	Se utiliza para los permisos que permiten el acceso a la ubicación del dispositivo.
MICROPHONE	Se utiliza para permisos que están asociados con el acceso de audio del micrófono del dispositivo.
PHONE	Se utiliza para los permisos que se asocian las funciones de telefonía.
SENSORS	Se utiliza para los permisos que se asocian las funciones de sensores. Ejemplo: lector de huellas dactilares, monitor de frecuencia cardíaca.
SMS	Se utiliza para los permisos de ejecución relacionados con los mensajes SMS de los usuarios.
STORAGE	Se utiliza para los permisos de ejecución relacionados con el almacenamiento externo compartido.

Tabla 2.3 – Permisos simplificados en Android

Los grupos de permisos están ordenados por criticidad o relevancia, mostrándose en la parte superior los más relevantes y terminando la lista con una categoría llamada “Otros” que engloba el resto de permisos solicitados y no agrupados previamente en

los permisos principales. Además, en esta categoría también se engloban los permisos propios que han definido apps de terceros.

La implementación de este nuevo modelo de permisos simplificados introduce nuevos riesgos de seguridad. Por ejemplo, una app que requiera disponer del permiso para leer SMS, antes solicitaba el permiso "READ_SMS", pero ahora solicitará el grupo de permisos "SMS", lo que permitirá que pueda acceder a todas las capacidades relacionadas con los SMS, como pueden ser el envío de SMS. Además, una aplicación que ya haya sido instalada y dispone de ciertos permisos, al recibir la siguiente actualización, obtendrá automáticamente acceso al grupo completo de permisos asociados a cada uno de los permisos individuales que ya disponía, lo que implica que obtendrá automáticamente acceso a muchos más permisos de los que se solicitaron inicialmente durante su instalación.

Algo muy preocupante también en cuanto a la seguridad se refiere es que esta actualización del modelo de permisos de Android otorga por defecto el permiso (existente previamente) de acceso completo a Internet a cualquier app. Esto es así ya que el permiso de acceso a internet ha sido eliminado de manera efectiva y aunque los desarrolladores de las apps tienen que seguir declarando que la aplicación en concreto requiere el permiso, los usuarios no verán la solicitud de ese permiso durante el proceso de instalación. Además, como hemos mencionado en el párrafo anterior las aplicaciones ya instaladas y que no disponían del permiso, lo podrán obtener automáticamente y sin la aprobación del usuario en la siguiente actualización[16].

2.3.3. PROCEDENCIA DE APLICACIONES

Con respecto a la implementación de las aplicaciones de Android, éstas tienen que estar firmadas digitalmente para que sean consideradas válidas y puedan ser distribuidas a través de Play Store y, posteriormente poder ser instalarlas en el dispositivo. Dicha firma se hace a través de un certificado, cuya clave privada será la del desarrollador de dicha aplicación. Android no realiza ninguna verificación adicional sobre la seguridad de la app o la firma digital, salvo comprobar la fecha de expiración del certificado asociado a la firma en el momento de la instalación de la app, permitiéndose certificados auto-firmados y no generados o validados por ninguna autoridad certificadora. Mediante la firma, únicamente se vincula (teóricamente) la aplicación a su desarrollador, asegura su integridad frente a modificaciones y se emplea para las actualizaciones de la app, verificando que el nuevo código proviene

del mismo desarrollador, es decir, del autor original de la aplicación[16].

2.3.4. VERIFICACIÓN DE APLICACIONES

Además de todo lo anterior, Google cuenta con Bouncer[19], que lleva a cabo un proceso de escaneo o verificación de seguridad automático sobre la apps que son publicadas por los desarrolladores a través de Play Store en busca de *malware.* Google Bouncer compara cada app con otras muestras de software malicioso conocidas, y también ejecuta las apps en un entorno virtual, simulando su ejecución en un dispositivo móvil Android, para analizar su comportamiento antes de proceder a su publicación oficial, e intentando identificar comportamientos maliciosos o anómalos asociados al *malware*. El sistema Bouncer analiza las apps nueva, apps ya existentes previamente en Play Store, reanaliza apps ya analizadas previamente una vez se incorporan nuevas capacidades y también analiza cuentas de desarrolladores de Android[16].

Por otro lado, desde la versión 4.2 de Android, el propio sistema dispone de capacidades para el escaneo o verificación de aplicaciones durante el proceso de instalación, con el objetivo de identificar y detectar la existencia de malware. Esta funcionalidad compara la app a instalar con una base de datos existente en Google - para ello el proceso de verificación envía a Google el nombre de la app, su tamaño, su hash SHA1, la versión y la URL asociada - con el objetivo de identificar si corresponde a una app conocida con comportamiento malicioso o dañino. En caso de identificarse como dañina, Android puede recomendar al usuario que no instale la app o incluso llegar a bloquear la app, no siendo posible proceder con su instalación[16].

Esta verificación se lleva a cabo con las apps que son instaladas manualmente por el usuario, ya sea desde una página web, un repositorio de apps no oficial o a través de ADB[20] (proceso conocido como *sideload*). Para las aplicaciones instaladas desde Play Store se confía en la verificación realizada por Google Bouncer. Hay que tener en cuenta que esta funcionalidad es necesario habilitarla previamente ya que por defecto viene deshabilitada en versiones anteriores a 4.4.4. Además, en las últimas versiones de Android, esta funcionalidad también comprueba y monitoriza las apps durante su ejecución[21] por si presentan algún comportamiento inesperado que no fue detectado durante la instalación. En caso de identificarse la app como dañina, Android puede recomendar al usuario que desinstale la app o incluso llegar a eliminarla de forma automática[16].

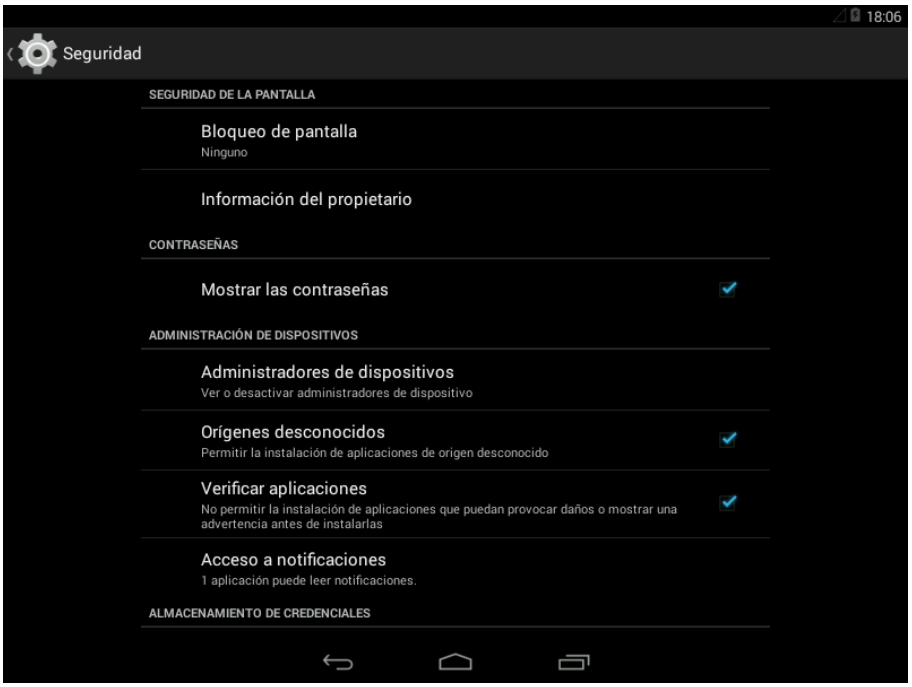


Figura 2.4 - Configuración Verificación de aplicaciones en Android 4.4

2.3.5. POLÍTICA PARA DESARROLLADORES DE PLAY STORE

En otro estado de cosas, Google ha reforzado su política para desarrolladores (si queremos leer toda la política la podemos encontrar en https://play.google.com/intl/ALL_es/about/developer-content-policy.html). A continuación listamos las principales novedades[22]:

- **Promoción de las aplicaciones:** Una aplicación publicada en Play Store no podrá participar en las siguientes actividades ni beneficiarse de ellas.
  - Promoción a través de publicidad engañosa.
  - Promoción o instalación de sistemas que provoquen un direccionamiento a Play Store o descargas de aplicaciones.
  - Promoción no solicitada a través de mensajes SMS.
- **Contenido sexual explícito:** están prohibidas las aplicaciones que promuevan la pornografía o incluyan material pornográfico. Además, Google mantendrá una política de tolerancia cero con las imágenes de abuso sexual infantil.
- **Aplicaciones peligrosas:**
  - No se permitirá la transmisión ni la inclusión de enlaces de virus, gusanos, defectos, troyanos o software malicioso.
  - Se prohíben las aplicaciones que recopilan información sin el conocimiento

del usuario: *software* espía, o que indique la ubicación del usuario.

- Una aplicación descargada de Play Store no puede modificar, sustituir o actualizar el código binario de su propio APK mediante ningún método distinto al mecanismo de actualización de Play Store.

- **Interferencia con el sistema:** Las aplicaciones y sus anuncios no deben añadir ni modificar marcadores ni opciones de configuración del navegador ni añadir accesos directos en la pantalla de inicio o iconos en el dispositivo como servicio a terceros o con fines publicitarios. Tampoco se permite que se muestren anuncios mediante las notificaciones del sistema ni deben animar, incentivar o engañar a los usuarios para que eliminen o inhabiliten aplicaciones de terceros.

2.3.6. ELIMINACIÓN REMOTA DE APLICACIONES

Este mecanismo fue diseñado como medida de seguridad en situaciones de emergencia para poder proteger a los usuarios en caso de que se produzca una distribución masiva de apps dañinas o maliciosas concretas. Además, permiten a Google, no sólo eliminar las aplicaciones de los dispositivos móviles, si no también instalar remotamente apps, característica que se emplea por ejemplo en la instalación de apps adquiridas desde el entorno web de Play Store[16].

Se sabe que Google hizo uso de esta funcionalidad en marzo de 2011[23] cuando se vieron afectadas unas 50 aplicaciones infectadas con malware (con el nombre de DroidDream) que fueron instaladas en más de 200.000 dispositivos móviles. Estas apps hacían uso de vulnerabilidades conocidas en versiones de Android previas a la versión 2.2.2, y específicamente, técnicas empleadas habitualmente para conseguir acceso como root a dispositivos móviles.

2.3.7. CIFRADO DE DATOS

Desde la versión 3.0, (orientada a tabletas) Android posee la capacidad para cifrar los datos almacenados en la memoria interna del dispositivo móvil de forma nativa. Además, también se pueden cifrar los datos de la tarjeta de almacenamiento externa (en aquellos dispositivos que puedan disponer de ella). La opción de cifrado se ha de habilitar, ya que por defecto viene deshabilitada y, concretamente en Android 4.x y posteriores es necesario establecer un código de acceso en el dispositivo móvil en forma de contraseña o PIN. A continuación mostramos, en la Figura 2.5, las pantallas para habilitar el cifrado en un dispositivo con Android 4.4.4.

Figura 2.5 – Pantallas de activación cifrado dispositivo

Hay que tener en cuenta que la implementación de cifrado del dispositivo, sólo cifra la partición o sistema de ficheros con los datos de usuario (userdata), correspondiente al punto de montaje "/data", y no toda la memoria del dispositivo. Otra consideración es que el proceso de cifrado es irreversible, es decir, si queremos que el dispositivo móvil vuelva a estar sin cifrar, hay que restaurar el dispositivo a sus ajustes de fábrica (eliminará todos los datos del terminal) y posteriormente restaurar los datos del usuario desde una copia de seguridad.

2.3.8. COMUNICACIONES SEGURAS

Android dispone de una implementación de los protocolos SSL/TLS , (con soporte para SSL v3.0 y hasta TLS v1.2). Al acceder a un sitio web mediante HTTPS, el dispositivo móvil intentará verificar el certificado digital asociado y su cadena de confianza, hasta la Autoridad Certificadora raíz reconocida correspondiente. En caso que no sea posible verificar la validez de un certificado, se generará un mensaje de advertencia,

indicando el motivo del error y permitiendo al usuario ver y analizar algunos detalles del motivo. Si por el contrario, se acepta el certificado digital, éste será válido hasta que se cierre la instancia, proceso o sesión actual del navegador web.

Por otra parte, desde la versión 4.0 de Android se soporta la utilización de mecanismos de autenticación web basados en certificados digitales cliente a través del navegador web existente por defecto, asimismo es posible utilizar los certificados cliente para la conexión a redes WIFI o redes VPN.

En cuanto a las comunicaciones seguras, Android (a partir de la versión 4.0) también proporciona soportes para las siguientes tecnologías y protocolos empleados por redes VPN.

- PPTP: Point-to-Point Tunneling Protocol (TCP/1723).
- L2TP/IPSec: Layer Two Tunneling Protocol (L2TP; UDP/1701) en combinación con Internet Protocol Security (IPSec; protocolos IP 50 y 51 + IKE: UDP/500).
    - L2TP/IPSec PSK
    - L2TP/IPSec RSA
- IPSec: Internet Protocol Security (protocolos IP 50 y 51 + IKE: UDP/500).
    - IPSec Xauth PSK
    - IPSec Xauth RSA
    - IPSec Hybrid RSA

La opción recomendada desde el punto de vista de seguridad es crear redes VPN basadas en IPSec o L2TP/IPSec.

2.3.9. CONTROL DE ACCESO FÍSICO

Como algo adicional al mundo lógico del dispositivo, Android permite fijar un PIN o código de acceso tanto en la tarjeta SIM como en el propio dispositivo móvil.

El PIN de la tarjeta SIM bloquea el acceso no autorizado a los servicios y capacidades de telefonía móvil. Si la tarjeta SIM está bloqueada solamente se permiten realizar llamadas a los servicios de emergencias.

Por otra parte, el código de acceso, ya sea PIN, contraseña, patrón de acceso, etc., del dispositivo móvil bloquea el acceso no autorizado al terminal, incluyendo sus datos, capacidades de comunicación y aplicaciones.

2.4. APLICACIONES EN ANDROID

Las aplicaciones en Android principalmente se desarrollan en Java, pero no corren sobre Java ME[24], si no sobre Dalvik, donde los códigos fuente se compilan a ficheros de bytecode .dex. Posteriormente los archivos .dex, el AndroidManifest.xml, todos los recursos, certificados y librerías propias de la aplicación son empaquetados en un archivo ZIP con la extensión .apk (Android Application Package). Toda aplicación hará uso de las distintas API's proporcionadas por Android, como hemos explicado en el punto 2.2) de forma que los componentes encargados de realizar cada tarea puedan ser manipulados sin problemas, asegurando la máxima flexibilidad.

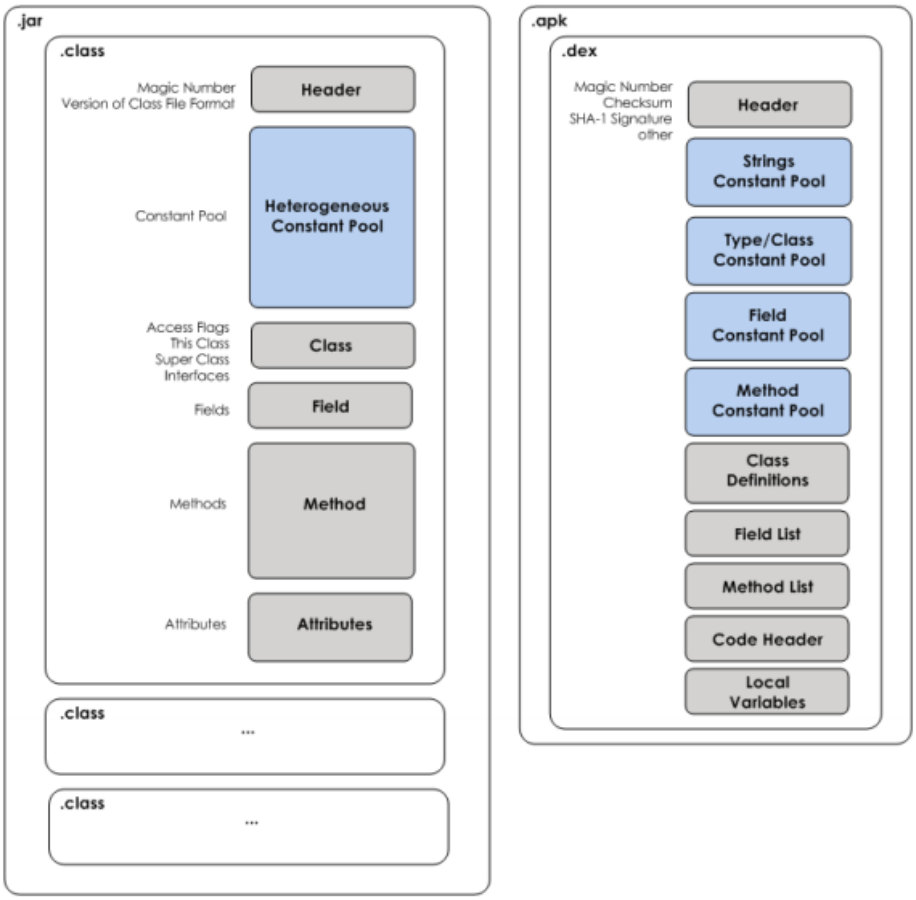


Figura 2.6 – Comparativa de la estructura interna de un archivo .class y un archivo .dex. Extraída de [d]

Los archivos .APK son los que nos permiten instalar una aplicación en el dispositivo y

están formados por los siguientes tipos de archivos[25]:

- **AndroidManifest.xml**: Ubicado en la raíz de la aplicación, este archivo es la definición de todas las características que tendrá la aplicación al ejecutarse en un dispositivo móvil. Es decir, contiene, los permisos, las escuchas, receptores, metadatos, versión, las versiones previas soportadas, las dimensiones de la pantalla, etc.
- **Classes.dex**: Este será el fichero compilado preparado para ejecutarse en la Máquina Virtual Dalvik.
- **Resources**: En esta carpeta encontramos todos los archivos externos que usamos para construir el proyecto, como por ejemplo iconos, audio, archivos planos de texto, los archivos .xml de diseño, etc.
- **Librerías aplicación**: El archivo .apk también contiene aquellas librerías de las cuales depende la aplicación.
- **Carpeta META-INF**: En ella se guardan archivos que corresponden a las Firmas Digitales de la aplicación. Con esta especificación se puede indicar quien es el creador y dueño de la aplicación, además debe contener el ID de desarrollador para ser reconocido y autenticado en procesos de comercialización.

Cuando ejecutamos una aplicación, ésta se asocia a un proceso único, que proporciona el entorno de ejecución de los componentes. De los cuales, uno es el componente inicial del programa. Cuando se ejecuta una aplicación se le asigna un proceso Linux y un único hilo de ejecución (thread), así todos sus componentes corren sobre el mismo proceso y thread.

Si analizamos la anatomía de una aplicación Android[26][27], podemos afirmar que está compuesta por sus *Activity*, las cuales, por lo general, son pantallas que permiten la implementación de las clases, los I*ntent* e I*ntenfilters* que son clases que permiten el desplazamiento entre pantallas, el *Broadcast Intent Receiver*, que es usado como un disparador de eventos frente a determinadas situaciones, los *Services*, que son códigos que se ejecutan en segundo plano, sin ninguna interfaz de usuario y el *Content Provider* que es el encargado de la gestión de datos. A continuación pasamos a detallar cada uno de estos componentes.

- ***Activity***: Es sin duda el componente más habitual de las aplicaciones para Android. Un componente *Activity* refleja una determinada actividad llevada a cabo por una aplicación, y lleva asociada típicamente una ventana o interfaz de usuario.

Dicho de otra forma, podemos decir que es una pantalla de la aplicación (aunque no contempla únicamente el aspecto gráfico). Cada actividad es implementada como una clase que extiende a la clase base. La clase mostrará una interfaz de usuario compuesta por *Views* que responderá a los distintos eventos que se produzcan. La mayoría de las aplicaciones están compuestas por varias pantallas, por lo que éstas, a su vez estarán compuestas por varias actividades. Las *Activities* deben estar declaradas en el archivo AndroidManifest.xml. Para poner un ejemplo, piénsese en una aplicación de mensajes de texto. En ella, la lista de contactos se muestra en una ventana. Mediante el despliegue de una segunda ventana, el usuario puede escribir el mensaje al contacto elegido, y en otra tercera puede repasar su historial de mensajes enviados o recibidos. Cada una de estas ventanas debería estar representada a través de un componente *Activity*, de forma que navegar de una ventana a otra implica lanzar una actividad o dormir otra. Android permite controlar por completo el ciclo de vida de los componentes *Activity*.

- ***Intent e IntentFilters***: Son clases especiales para moverse de una pantalla a otra. Describe lo que una aplicación quiere hacer, o dicho de otro modo, un I*ntent* consiste básicamente en la voluntad de realizar alguna acción, generalmente asociada a unos datos. Lo más importante de esta estructura es la acción y los datos para llevarla a cabo. Lanzado un *Intent*, una aplicación puede delegar el trabajo en otra, de forma que el sistema se encarga de buscar qué aplicación de entre las instaladas, es la que puede llevar a cabo la acción solicitada. Los *Intents* están incluidos en el AndroidManifest.xml ya que describen dónde y cuándo puede comenzar una actividad.
- ***Broadcast Intent Receiver***: Se utilizan para lanzar alguna ejecución dentro de la aplicación actual cuando un determinado evento se produzca (generalmente, abrir un componente *Activity*).Por ejemplo, una llamada entrante o un SMS recibido. Este componente no tiene interfaz de usuario asociada pero puede usar Notification Manager para avisar al usuario de que algo ha pasado. Para que *Broadcast Intent Receiver* funcione, no es necesario que la aplicación en cuestión sea la aplicación activa en el momento de producirse el evento. El sistema lanzará la aplicación si es necesario cuando el evento monitorizado tenga lugar.
- ***Service***: Es un código que se ejecuta sin ninguna interfaz de usuario, en segundo plano, para permitir así la ejecución de otras actividades. Un ejemplo típico de este componente es un reproductor de música. Es importante notar que es posible conectarse a un servicio mediante la interfaz que la actividad pone a disposición

del usuario.

- ***Content Provider***: Este componente permite que cualquier aplicación pueda almacenar sus datos en ficheros, bases de datos (SQLite), etc. Tienen sentido si se quiere compartir datos con otras aplicaciones. Es un servicio que da a las aplicaciones, capacidad de comunicación bilateral con otras aplicaciones de forma interna.

Conviene remarcar que no todas las aplicaciones tienen los cinco componentes, pero cualquier aplicación será una combinación de éstos.

2.5. EL MALWARE EN ANDROID

Android es el sistema operativo más implantado en dispositivos comprados (incluyendo pcs, portátiles, tabletas, etc.) y, la proyección para el año 2017, es que va a continuar siéndolo como podemos ver en la Figura 2.7 extraída de un informe de Gartner[28].

Worldwide Devices Shipments by Operating System (Thousands of Units)

Operating System	2012	2013	2014	2017
Android	497,082	860,937	1,069,503	1,468,619
Windows	346,457	354,410	397,533	570,937
iOS/MacOS	212,899	293,428	359,483	504,147
RIM	34,722	31,253	27,150	24,121
Others	1,122,213	871,718	702,786	396,959
Total	**2,213,373**	**2,411,796**	**2,556,455**	**2,964,783**

Note: Devices include notebooks and desk-based PCs, tablets, ultramobiles and mobile phones.

Source: Gartner (April 2013)

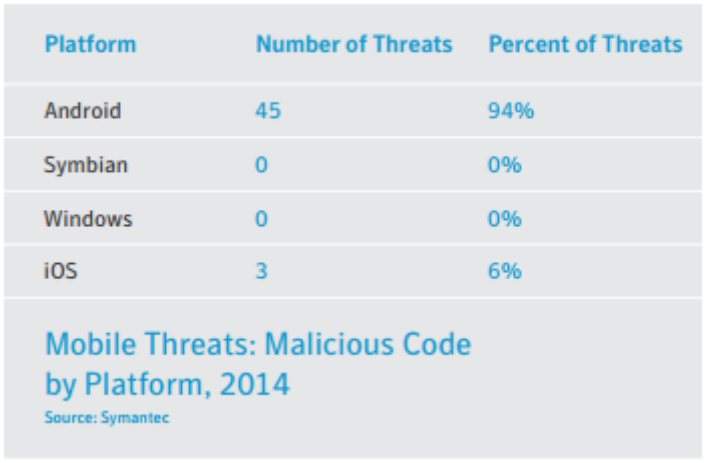
Figura 2.7 – Dispositivos por Sistema Operativo en el mundo. Fuente Gartner [e]

Como reverso de la misma moneda, este hecho propicia que esta plataforma sea muy atractiva para los desarrolladores de malware, ya que con un desarrollo pueden llegar a un número enorme de potenciales víctimas. Este hecho lo constatan informes de diferentes compañías de protección de antivirus, como pueden ser:

- **F-Secure**: En varios informes sobre el estado de las amenazas a dispositivos móviles, esta empresa confirma que Android es (y de forma destacada), la plataforma dónde más malware se desarrolla. Por ejemplo, en los nueve primeros

meses del año 2013 detectaron 633 nuevas familias y/o variantes de malware, de éstas, 610 fueron creadas para la plataforma Android[29]. En el primer trimestre de 2014 de 277 nuevas familias y/o variantes 275 fueron para Android[30].

- **Fortinet**: En el año 2013 el 96,5% del malware dirigido hacia dispositivos móviles fue creado para esa plataforma[31].
- **Juniper**: En el año 2011 el malware para Android representaba el 46,7% del total, cifra que en el año 2013 se dobló hasta llegar al 92%[32].
- **Symantec**: En el año 2014 descubrieron 48 nuevas amenazas para dispositivos móviles, de las cuales 45 (es decir, un 94%) estaban desarrolladas para la plataforma Android[33].



Platform	Number of Threats	Percent of Threats
Android	45	94%
Symbian	0	0%
Windows	0	0%
iOS	3	6%

Mobile Threats: Malicious Code by Platform, 2014
Source: Symantec

Figura 2.8 – Malware detectado en 2014 según S.O. Fuente Symantec [f].

Por otra parte, según el CCN-CERT, en los próximos años se duplicará el número de amenazas a Android y las vulnerabilidades en los dispositivos móviles, plataformas y aplicaciones. Los datos comprometidos se usarán para otros ataques o para su venta en el mercado negro[34].

A continuación, vamos a clarificar conceptos y establecer definiciones para seguir posteriormente describiendo las tipologías de malware según su carga útil y terminar analizando casos de malware en Android según diferentes actores del escenario de la seguridad.

2.5.1. CONCEPTOS Y DEFINICIONES

A continuación se definen los conceptos y definiciones fundamentales relacionadas con el malware para proporcionar al lector la terminología que se utilizará a los largo

del Trabajo Fin de Master (TFM). Una parte importante de este apartado ha sido extraída de la "Tesis: Nuevo enfoque para la detección de malware basado en métodos de recuperación de información"[35].

- **Malware**: El término malware es la combinación de las voces inglesas *malicious* (malicioso) y *software*. Se refiere a cualquier programa informático diseñado con la intención de dañar ordenadores, redes o información. Si centramos su definición en la forma en que el malware suele comportarse podemos decir que el malware es: "Cualquier trozo de código añadido, cambiado o borrado de un sistema software para causar intencionalmente daño o subvertir la función del sistema"[36].

Actualmente se utiliza el término malware para nombrar, en el sentido amplio de la palabra, amenazas, pero en el pasado la forma más común de nombrarlas era la expresión virus informático.

Teniendo en cuenta que el término malware se refiere a cualquier tipo de software con intenciones maliciosas, podemos distinguir diversos tipos de malware incluidos en esta definición.

- **Virus**: "Un virus es un programa que se replica infectando a otro programa; un sector de arranque o de partición; o un documento que permita macros. Normalmente, añade una copia de sí mismo a los ficheros de la víctima"[37].
- **Gusano**: Muy similares a los virus en cuanto a que se auto-replican, se diferencian de ellos en que no necesitan añadirse a un programa existente. Así, según los define M. Dalla Preda[38] son: "Un programa malicioso que utiliza la red para enviar copias de sí mismo a otros sistemas se denomina gusano. Al contrario que los virus, los gusanos no necesitan que su portador les lleve a otro sistema, ellos mismos se encargan de propagarse a través de la red (...)".
- **Caballo de Troya**: También llamados troyanos son un tipo de software no-auto-replicante. "Como los virus, los caballos de Troya se ocultan dentro de programas que pueden parecer útiles o, al menos, inofensivos para un usuario. Los caballos de Troya pueden ser tanto programas legítimos corruptos que ejecutan código malicioso al ponerse en funcionamiento; como programas que ejecutan código malicioso directamente, enmascarados como otra cosa, para conseguir el descuido del usuario y, así, alcanzar la complicidad necesaria para llevar a cabo sus objetivos"[38].
- **Spyware**: Un programa de spyware es un tipo de software que envía información

personal recogida del sistema víctima a una tercera parte sin el consentimiento del usuario del dispositivo[37][42].

- **Puerta trasera**: "Una puerta trasera es un programa informático diseñado para superar las políticas de seguridad con el fin de permitir a entidades externas tener control sobre una máquina o una red de forma remota. Las puertas traseras pueden ser programas únicos o alojarse dentro de versiones corruptas de programas benignos." [38]. A esta definición le podemos agregar que las puertas traseras habilitan un método para acceder a un sistema, proporcionando una conexión remota a hackers o a otro malware.
- **Rootkits**: Ataques sofisticados que modifican ficheros y/o librerías del S.O. para ocultar su existencia (o de otros códigos maliciosos, suelen ser puertas traseras). El proceso de instalación puede ser de forma automática o un atacante podría instalarlo una vez que ha obtenido permisos de root. Permite obtener el control del equipo y para ello falsea las llamadas al sistema. Hay dos tipos de rootkits, en modo kernel o usuario.
- **Bomba lógica**: "Una bomba lógica es un malware cuyo cuerpo se activa en una situación particular, en un instante concreto de tiempo o cuando ciertas condiciones se satisfacen"[37]. Los troyanos que se activan en fechas concretas se denominan bombas de tiempo.
- **Bot**: "Un bot es un tipo de programa malicioso que permite a un atacante tomar el control del equipo infectado. A su vez, participa en un sistema de control a gran escala de máquinas víctima. Las máquinas infectadas con el programa bot están a la espera de recibir órdenes y forman una botnet o red de bots, es decir, una red de máquinas controladas fácilmente al unísono por un solo atacante"[37].
- **Phishing**: Denomina a las acciones desarrolladas que se cometen mediante el uso de algún tipo de ingeniería social y/o malware, con el fin de obtener información confidencial para acceder de manera no autorizada a determinados servicios de forma fraudulenta. Es decir, se pretende obtener acceso a servicios suplantando en ellos la identidad de su legítimo titular. El ciclo de phishing consta de 3 fases claramente diferenciadas: Captura de datos, captación de mulas y finalmente transferencia, monetización y envío de lo defraudado[39].
- **Rooting**: Podría no ser considerado un tipo de malware en sí mismo ya que usuarios legítimos pueden utilizar este software para obtener control privilegiado sobre el dispositivo y, por ejemplo reemplazar el sistema operativo desde una versión más antigua a una más nueva. Desde otra perspectiva, Google lo considera malware ya que permite a quien lo hace escalar privilegios y saltarse los

niveles de seguridad establecidos por el fabricante.

Hay que tener en cuenta que el malware actual no siempre se puede clasificar fácilmente en estas categorías debido a que en ocasiones se combinan más de una de las características que hemos presentado en un mismo espécimen.

Otras consideraciones a tener en cuenta cuando hablamos de malware son las técnicas de ofuscación de código, que se utilizan con la finalidad de ocultar el comportamiento real de sus ejecutables. Como concepto básico podemos decir que un ofuscador es un programa informático que realiza ciertas transformaciones a un ejecutable. El código resultante es equivalente, en cuanto a funcionalidad, al código original pero más difícil de comprender. Hay diferentes técnicas, que tampoco consideramos que es necesario entrar en detalle pero sí podemos enumerar: transformaciones de capa, ofuscaciones de datos y transformaciones de control.

El malware ofuscado, resultante de aplicar las técnicas mencionadas anteriormente, se conoce como malware metamórfico, cuyo objetivo principal es cambiar la apariencia del código malicioso manteniendo intacta su funcionalidad. Lo que implica que el malware metamórfico se basa en una funcionalidad maliciosa unido a un motor metamórfico. Este motor cambia su código en cada infección del ejecutable.

También existe el malware polimórfico que consiste en que el cuerpo real de un ejecutable malicioso se oculta (o empaqueta) para suprimir la posibilidad de encajar con ninguna firma conocida. Los empaquetadores son aplicaciones informáticas que ocultan un ejecutable reduciendo su tamaño y su comprensibilidad.

Para terminar con este apartado y apoyándonos en lo acabado de explicar pasaremos a definir:

- **Variante de malware**: “Una variante de malware es el resultado de aplicar cualquier transformación que se pueda realizar a un ejecutable malicioso. Estas transformaciones incluyen transformaciones de ofuscación y métodos de empaquetado que mantienen la funcionalidad del código malicioso intacta”[39]. Es decir, es el nuevo malware creado mediante el uso de diferentes técnicas de ofuscación.
- **Familia de malware**: “Definimos una familia de malware como el conjunto de diferentes mutaciones de un ejecutable original y de todas las variantes con la misma funcionalidad”[40]. De esta manera podemos decir que una familia de

malware se compone por diversas variantes de malware, que son las versiones ofuscadas (metamórficas y polimórficas) de una funcionalidad original. En el **Anexo I** presentamos una tabla con familias de malware actualmente existentes[41].

En este apartado vamos a caracterizar el malware desde la perspectiva de la carga maliciosa que incorporan[42].

2.5.2. FINES DEL MALWARE

- **Control remoto**: La mayoría de la carga maliciosa del malware tiene la característica de convertir los teléfonos en bots para controlarlos remotamente. Este control remoto, se intenta hacer principalmente usando tráfico web basado en HTTP para recibir los comandos del servidor de C&C. En otro aspecto, para dificultar la localización del atacante y lo que se pretende, hay familias de malware que cifran las URLs del servidor remoto de C&C, así como la comunicación con él.
- **Generar gastos monetarios**: Una forma muy típica de generar estos gastos es que se suscriba al usuario infectado a los servicios SMS Premium. En Android hay un permiso llamado "sendTextMessage" que permite enviar mensajes SMS en segundo plano sin conocimiento del usuario. Otra forma de generar gastos podría ser hacer llamadas telefónicas, también en segundo plano.
- **Robo de información personal**: La información que puede interesar al atacante puede ser mensajes SMS, números de teléfono, cuentas de usuario, contraseñas de servicios, direcciones de correo electrónico, etc.
- **Pago por rescate de información (Ransomware)**: Se cifran determinadas partes o archivos del sistema infectado y se pide un rescate (monetario) a cambio de quitar esta restricción. Normalmente un ransomware se transmite tanto como un troyano como un gusano, infectando el sistema operativo, por ejemplo, con un archivo descargado o explotando una vulnerabilidad de software.

2.5.3. CASOS DE MALWARE EN ANDROID.

Según el informe de Google "Android Security 2014 Year in Review"[43] menos del 1% de los dispositivos tenían algún tipo de malware (ellos lo denominan Aplicaciones Potencialmente Peligrosas – PHA-) instalado y, centrándose en los dispositivos que solamente descargan aplicaciones de Play Store el porcentaje desciende hasta un 0,15%. Estos porcentajes son obtenidos como media a nivel global aunque distinguen que, por países, esto puede variar. A continuación mostramos en la Figura 2.9, una

tabla obtenida del citado informe dónde se puede apreciar, durante el año 2014, los países con más casos de dispositivos infectados y la media a nivel mundial. Esta tabla muestra los casos de aplicaciones obtenidas desde otro repositorio diferente del Play Store.

Locale	Q1	Q2	Q3	Q4	2014 Average
JP	0.0919%	0.0688%	0.0457%	0.0742%	**0.0702%**
IR	0.1810%	0.1157%	0.0774%	0.1802%	**0.1386%**
KR	0.1938%	0.1448%	0.1630%	0.2513%	**0.1882%**
CN	0.3097%	0.5006%	0.5137%	0.5145%	**0.4596%**
US	1.0069%	0.5678%	0.3376%	0.4889%	**0.6003%**
GB	1.5525%	0.6851%	0.2836%	0.3143%	**0.7089%**
Worldwide	1.0590%	1.0625%	0.4679%	0.5670%	**0.7891%**
BR	1.5518%	1.0639%	0.4989%	0.8836%	**0.9996%**
ID	2.0603%	1.1477%	0.5928%	0.7520%	**1.1382%**
AE	3.0692%	1.7243%	0.5016%	0.5859%	**1.4703%**
RU	3.2671%	8.2968%	1.7496%	2.1057%	**3.8548%**

Figura 2.9 – Casos de malware en Android, por países. Fuente Google [g]

Otra gráfica, que consideramos interesante mostrar es la Figura 2.10 (obtenida del mismo informe), donde se nos ofrece visibilidad de los casos de infecciones haciendo distinción de los diferentes tipos de malware. Como en el caso anterior Google se centra en las aplicaciones instaladas desde otros repositorios que no son Play Store. Si hacemos un análisis de la misma, podemos ver que durante la primera mitad del año, el malware más común fue el Spyware (se puede observar cómo ha decrecido a lo largo del año). El segundo tipo más común fue el malware genérico. Esta categoría se considera como tal cuando las aplicaciones se sabe que son potencialmente dañinas basándose en la asociación con anteriores que se han identificado como dañinas, en particular, estas apps estaban siendo descargadas de sitios web que tenían como objetivos dispositivos rusos para cometer fraude con SMS y WAP. En la segunda mitad del año el tipo más común ha sido rootear el dispositivo.

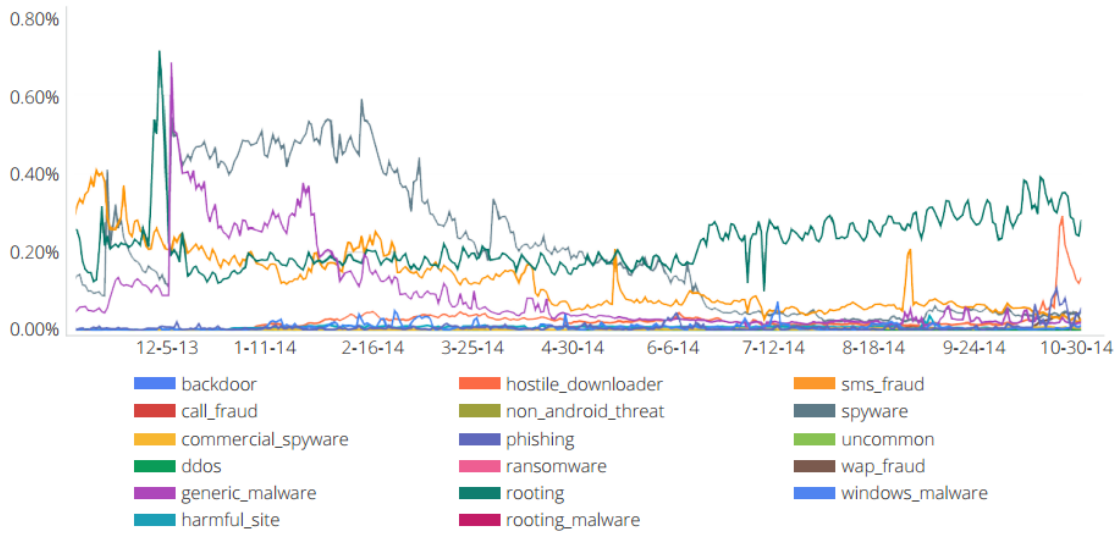


Figura 2.10 – Casos de malware en Android, por tipos. Fuente Google [g]

Si analizamos otro informe del mismo periodo (en este caso de la compañía Symantec "Internet Security Threat Report 20"[44]) vemos que las cifras difieren respecto a las de Google de forma evidente. Por una parte, como se muestra en la Figura 2.11, en cuanto a aplicaciones clasificadas como malware, del total de apps analizadas, mientras que Google dice que éstas sólo representan menos del 1%, en este informe vemos que, en el año 2014, son el 16% aproximadamente y que de 6,3 millones de apps analizadas 2,3 millones son programas que no contienen virus u otros componentes maliciosos pero sí que pueden alojar elementos peligrosos para el usuario, (lo que se conoce como Grayware). Por ejemplo, herramientas de hacking, spyware, adware, etc.). Además, 1,3 millones de apps contienen Madware, que son técnicas agresivas para presentar publicidad que obviamente los usuarios no quieren tener instaladas.

Esta considerable cantidad de malware, vemos, en la misma tabla como en años anteriores, representaba un porcentaje menor que en el año 2014 (un 11,5% en el año 2013 y un 7% en el año 2012) pero en ningún caso llegan a los ínfimos niveles que asegura Google.

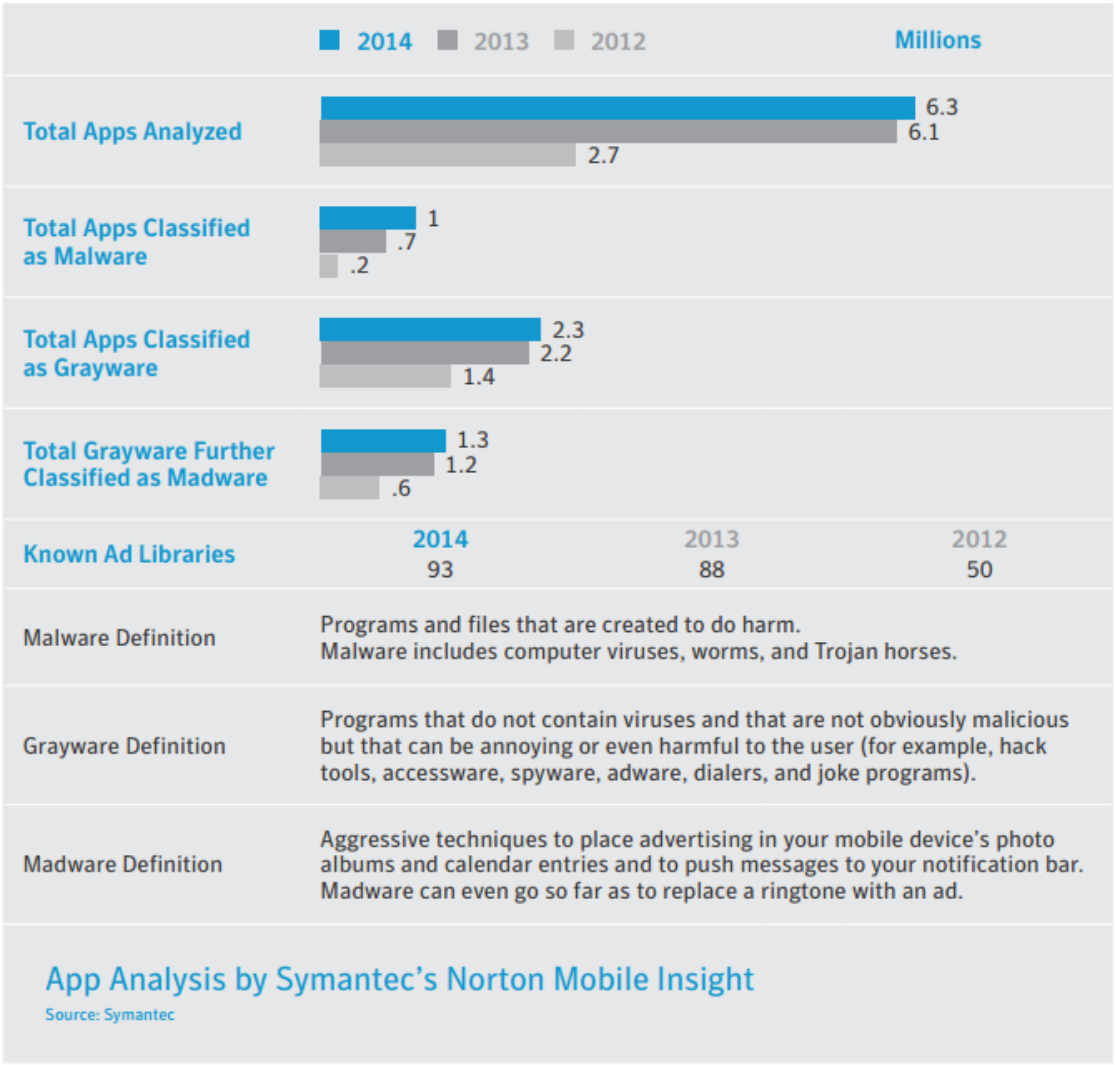


Figura 2.11 – Casos de malware en Android. Fuente [h].

En cuanto a los diferentes tipos de malware que se han dado en el periodo 2012-2014, según Symantec, éstos han sido principalmente los que presentamos en la Figura 2.12.

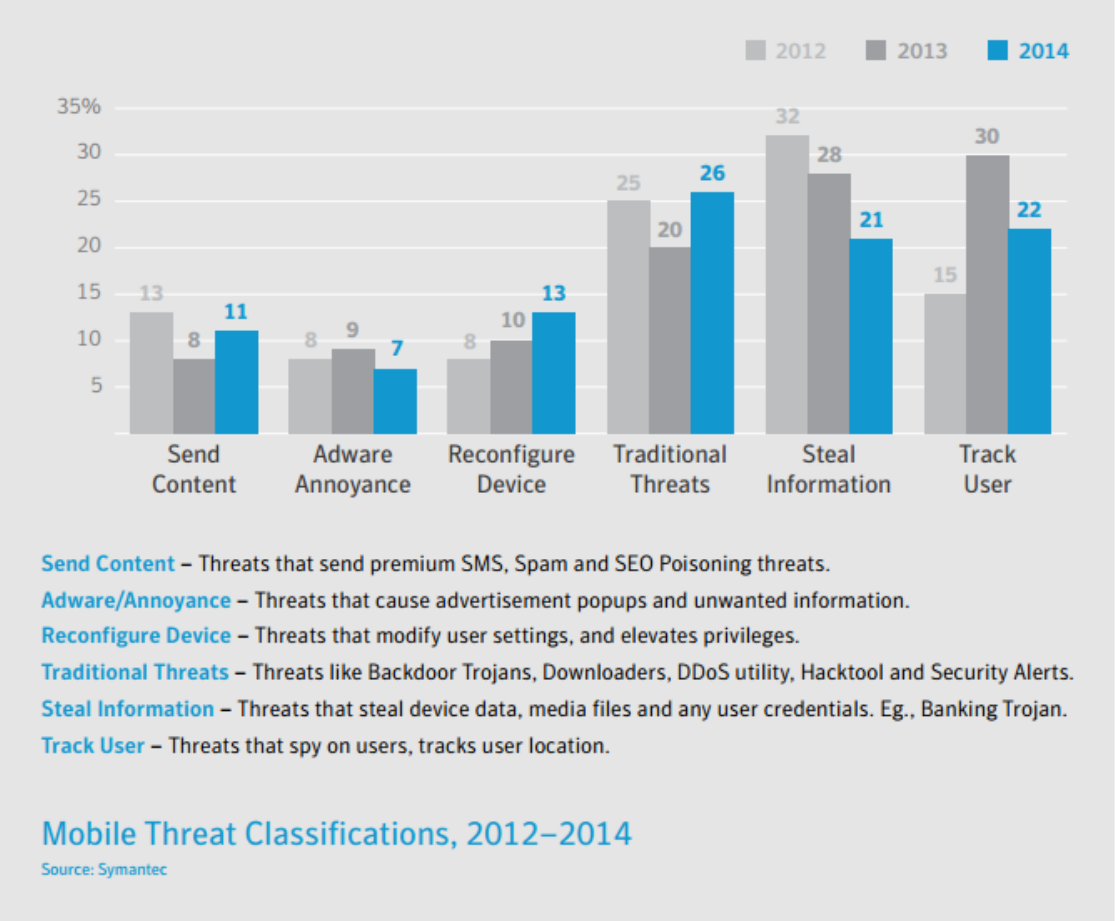


Figura 2.12 – Casos de malware en Android por tipos. Fuente [h].

Si consideramos la divergencia de resultados, sobretodo en el caso de apps con malware, podemos pensar (y no sería de extrañar) que hay intereses comerciales detrás de los resultados presentados, debido a que, por una parte a Google le interesa decir (y mostrar) que su plataforma es segura y "os enseño" que hay muy poco malware (gracias en parte a todas las medidas de seguridad desplegadas por la compañía) pero por otra parte, Symantec, como proveedor de servicios de seguridad le interesa vender su producto, por tanto debe "enseñar" que hay el suficiente malware en las apps para que su solución sea necesaria y "vendible". En el punto 3.7 entraremos más en detalle en este debate sobre si realmente hay tanto malware (que afecte a los dispositivos) para que realmente sea necesario tener (o comprar) una herramienta de prevención de malware.

2.6. FORMAS DE INFECCIÓN

Hay principalmente 3 técnicas de ingeniería social para la instalación del malware. Hay que tener en cuenta que estas técnicas no son mutuamente excluyentes entre ellas ya que diferentes variantes del mismo tipo de malware pueden utilizar diferentes técnicas para atraer a los usuarios para su descarga.

- **Re-empaquetado**: Los autores de malware localizan y descargan aplicaciones populares que son lícitas. Las desensamblan, le incorporan la carga dañina, las vuelven a ensamblar y cargan la nueva app en Androids Markets, el oficial o alternativos. El usuario, será vulnerable en cuanto se descargue e instale la aplicación infectada.
- **Instalación al actualizar**: Esta técnica es más difícil de detectar. También consiste en re-ensamblar una app lícita, pero en lugar de incorporar la carga dañina sólo se incluye un componente de actualización que obtendrá o descargará la carga dañina en la ejecución. Como resultado un escaneado de la app instalada no encontrará nada dañino y la considerará válida. Hay dos variantes de esta técnica. La primera hará la instalación de la carga dañina en modo silencioso, es decir, sin la intervención del usuario. La segunda hará que cuando se ejecute la aplicación, le aparecerá un diálogo de actualización al usuario diciendo que hay una nueva versión disponible, entonces, será el propio usuario el que se descargará la actualización con la carga dañina.
- **Descarga directa**: Lo que hacen esencialmente es que se ofrece al usuario descargar apps que tienen aspecto de “interesantes”. Por ejemplo, un tipo de malware puede empezar con un anuncio en una app lícita, donde le ofrecen que se puede descargar una app de análisis del uso de la batería para mejorar su uso. Este anuncio, si se enlaza, enviará al usuario a un sitio web malicioso o directamente a un Android Market falso desde donde se descargará la app que no hará lo que se le prometió, si no que puede suscribir al usuario a un servicio de envío de SMS Premium sin el conocimiento del usuario.

Además de las explicadas hasta ahora existen otras técnicas, que podríamos decir que es el propio malware el que se intenta instalar:

- El primer grupo es considerado software espía (spyware). Eso explica probablemente por qué los atacantes no tienen motivaciones o la necesidad de atraer a la víctima para su instalación.

- El segundo grupo incluye a aquellas aplicaciones falsas que se hacen pasar por aplicaciones legítimas, pero sigilosamente realizan acciones maliciosas, como el robo de credenciales de los usuarios o el envío de mensajes en segundo plano.
- El tercer grupo contiene aplicaciones que también incluyen intencionalmente funcionalidades maliciosas (por ejemplo, el envío de mensajes SMS no autorizadas o suscribirse a algún servicio de valor añadido de forma automática). Pero la diferencia con el segundo grupo es que no son falsas. En su lugar, pueden proporcionar la funcionalidad que afirmaban. Pero de forma oculta para los usuarios, también incluyen ciertas funcionalidades maliciosas.
- El último grupo incluye aquellas aplicaciones que se basan en el privilegio de root para funcionar bien. Sin embargo, no preguntan al usuario para conceder el privilegio root para estas aplicaciones, ellos aprovechan conocidos exploits de root para escapar del entorno limitado de seguridad incorporado. Aunque estas aplicaciones no pueden demostrar claramente las intenciones maliciosas, el hecho de utilizar los privilegios de root sin permiso del usuario parece cruzar la línea.

2.7. HERRAMIENTAS DE PREVENCIÓN

En este apartado, antes de nada, nos planteamos si realmente es necesario tener una herramienta de prevención de malware instalada en Android debido a que hay opiniones y acciones que apuntan en ambos sentidos. Seguidamente veremos una clasificación de las herramientas, en cuanto a cómo trabajan. Finalmente mostraremos un listado (no exhaustivo) de las herramientas que actualmente hay desarrolladas señalando las que son de pago y las que no.

2.7.1. ¿ES NECESARIO DISPONER DE ALGUNA HERRAMIENTA DE PREVENCIÓN?

Para comenzar haremos un pequeño inciso respecto a las historia de estas herramientas en Android. A pesar que la plataforma tiene poco tiempo (el primer dispositivo comercializado con Android fue en octubre de 2008) ya hay antivirus que datan de esas fechas, concretamente el primer antivirus que salió al mercado fue anunciado en noviembre de 2008 y lo desarrolló la empresa SMobile Systems, se llamó Smobile VirusGuard for Android, tenía un coste de 9,99$ y servía, según la empresa, para “proteger a tu G1 del ataque y proliferación de virus para móvil, spyware y otro malware”[45]. Suponemos que, como ya habían surgido virus para otras

plataformas móviles (Symbian, Palm, Pocket PC)[46] y además, este nuevo entorno, en manos de Google, iba a desarrollarse con fuerza, todo apuntaba a que sería un buen negocio desarrollar aplicaciones de protección.

Con el tiempo, y cada vez más, aparecieron nuevas empresas, o empresas que ya se dedicaban a la seguridad informática enfocaron (o ampliaron) su negocio hacia la protección antivirus en dispositivos móviles. Algunas de ellas muy conocidas como McAfee (ahora Intel Security), Norton, Symantec, Kaspersky, Avast, Panda Security, etc. y otras no tan conocidas, como pueden ser Hornet Antivirus, Zoner Antivirus, ESET, etc. Así pues, hoy en día la cantidad de empresas que desarrollan herramientas de protección para la plataforma Android es muy amplia, habiendo tanto apps gratuitas como de pago. Con estos datos, podemos pensar que si hay tantas empresas desarrollando apps en torno a la seguridad de la plataforma Android es porque realmente será necesaria (además que seguramente será un buen negocio).

Por otra parte, hay voces discordantes en cuanto a que realmente sean necesarias este tipo de aplicaciones teniendo en cuenta la cantidad de medidas de seguridad que Google ha diseñado en el despliegue de aplicaciones que finalmente el usuario puede instalarse (explicadas en el punto 2.3 del TFM). Hemos leído, (en julio de 2014) que el ingeniero jefe de seguridad de Android en Google declaró “No creo que más del 99% de los usuarios obtengan reales beneficios al usar estos antivirus. Ciertamente no hay razón por la que se deban instalar estas aplicaciones, ya que Android ya dispone de la seguridad necesaria desde fábrica. Si mi trabajo lo requiriera, este tipo de protección tendría sentido, pero creo que el usuario promedio de Android no requiere instalar antivirus porque simplemente no sirven para nada”[47]. Hay que puntualizar que recomienda estar a la última versión de Android (cosa que no siempre es posible, como hemos visto, hay mucha fragmentación) e instalarse las apps siempre desde el Play Store, algo que no todo el mundo hace. Además, de no rootear el dispositivo ya que entonces el modelo de seguridad implementado queda seriamente comprometido.

Si seguimos investigando sobre la cuestión en internet, la mayoría de las opiniones (de sitios web especializados en Android) son del tipo “no, pero”, es decir, en principio no es necesario debido a:

- La mayoría de usuarios hacen un uso ordinario de los dispositivos, esto puede ser, navegar, leer el correo, mensajería online, descargarse apps del Play Store, etc. y, sí que es cierto que Google ha implementado muchos mecanismos para proteger

el sistema, con lo que podemos estar bastante tranquilos si no vamos a descargar apps de orígenes no confiables y utilizamos el sentido común al tratar la información que nos llega por las diferentes vías.

- El antivirus consume recursos tanto de memoria, como de CPU, lo que implica pérdida de rendimiento (y no todos los usuarios tienen un dispositivo de última generación y/o gama alta) o de batería (reducción de autonomía)[48].
- Los antivirus no son infalibles[49].
- Según el Ingeniero jefe de seguridad de Android "La mayoría de personas nunca verán una aplicación potencialmente dañina, ni conoce a nadie que haya instalado una. Creo que se exagera con el nivel de riesgo de Android"[50].

Y en cuanto al "pero" (es necesario/útil tenerlo instalado) de las opiniones, éstas se basan en:

- Dependerá del perfil y los conocimientos del usuario. Es decir, si se le da un uso intensivo al móvil, es posible que se acabe navegando por sitios no siempre confiables, se descarguen apps fuera del Play Store o contenidos que no se conoce el origen de los mismos, se rotee el dispositivo, etc.[48][51]
- Google compró en el año 2012 VirusTotal[52] (el analizador de ficheros y enlaces web sospechosos), referencia mundial en cuanto al análisis de malware. Pues bien, Virustotal también cuenta con una aplicación para Android que simplifica el proceso de escaneado de las apps instaladas en el dispositivo. Realmente no es un antivirus pero presenta informes al usuario sobre lo que los antivirus encuentran en ellas[53].
- Los virus son algo real en los dispositivos móviles, y sobre todo en Android, que es el sistema operativo más popular[54].
- Si el dispositivo va sobrado de recursos, el usuario no va a verse afectado por la herramienta de protección, lo que no le hará ningún daño.
- La mayoría de herramientas son suites de seguridad y permiten como funcionalidades adicionales a la protección contra malware, por ejemplo: localizar el dispositivo si se extravía, borrar remotamente los datos, herramientas de backup y restauración, etc.[55]

Con esta información, la polémica está servida y volvemos a la cuestión inicial. ¿Es necesario disponer de alguna herramienta de prevención? Nuestra opinión personal es que sí, ya que no siempre se siguen las mejores prácticas de seguridad (que

explicaremos en el Capítulo 7) y las amenazas están ahí.

2.7.2. CARACTERÍSTICAS GENERALES DE HERRAMIENTAS DE PREVENCIÓN

Empezaremos diciendo que las herramientas de prevención de malware son programas cuya finalidad es prevenir y evitar la infección de malware, impidiendo también su propagación. Podemos generalizar diciendo que los antivirus tienen tres componentes principales[56]:

- **Vacuna o motor antivirus**: Programa que actúa en tiempo real, analizando los archivos que son abiertos o los programas que se ejecutan.
- **Motor de detección**: Programa cuya función es realizar el escaneo de los archivos, directorios o unidades que se seleccionan.
- **Desinfectador**: Programa que una vez localizado el virus y desactivada su estructura procede a eliminarlo, reparando sus efectos en el sistema (esto último no siempre es posible).

Por otra parte, en cuanto a la forma de análisis, las herramientas se pueden dividir en dos categorías diferentes[57]:

- **Detector sintáctico de malware o método clásico de firmas**: También conocidos como métodos estáticos, buscan ficheros para encontrar patrones de bytes sospechosos que se han almacenado anteriormente en una base de datos. Lo que se conoce como una firma, no es más que un extracto o una función resumen del cuerpo del malware. Esta aproximación tiene la limitación, en cuanto a su efectividad, que sólo detectará ejecutables maliciosos conocidos previamente, por lo que variantes relativamente simples de malware pueden pasar desapercibidos sin problemas. Por el contrario, la mayor ventaja de este método es que, debido a que las técnicas de comparación de patrones tienen una complejidad controlada, son realmente rápidos y tienen una tasa de falsos positivos baja.
- **Detección basada en el comportamiento**: Se basa en complejas estructuras e información estadística para representar un comportamiento, es decir, realizan un análisis dinámico. El enfoque consiste en modelar y detectar los comportamientos sospechosos que un malware puede generar para así poder identificarlos. También se podría haber optado por modelar el comportamiento del software benigno y así, medir las desviaciones de estos modelos, pero esta segunda

aproximación favorecería los falsos positivos (software legítimo incorrectamente catalogado como malicioso).

Si nos centramos en las funcionalidades que ofrecen, como ya se ha mencionado antes, la mayoría de herramientas han evolucionado hacia suites de seguridad y permiten, entre otros:

- **Protección antimalware**: La funcionalidad que más nos interesa y que todas tienen por defecto. Protege al dispositivo contra apps maliciosas. Hay diferentes modos de realizar la detección, en tiempo real, bajo demanda o de forma planificada.
- **Protección antirrobo/pérdida**: En caso de pérdida o robo, el software ofrece una serie de funciones que ayudarán al propietario legítimo del dispositivo a encontrar y/o proteger sus datos. Estamos hablando de: geolocalización, bloqueo del terminal, borrado remoto, que suene una alarma en el dispositivo, si la persona que no es propietaria consigue acceder al sistema, notificación cambio de tarjeta SIM, hacer una foto con la cámara delantera y enviarla a una dirección de correo previamente configurada por el propietario, protección contra desinstalación.
- **Protección de la privacidad**: La idea es proteger los datos personales y confidenciales de aplicaciones peligrosas y accesos ajenos. Permite por ejemplo: perfiles de aplicaciones de varios usuarios (se controla qué pueden ver los demás en el dispositivo), alertas sobre aplicaciones (que intentan acceder a datos que no requieren), bloqueo de aplicaciones (mediante PIN), etc.
- **Navegación segura**: Cada URL a la que se pretende acceder es previamente chequeada con los sistemas en la nube del fabricante y, en caso de que no sea de confianza avisa al usuario que no es la mejor idea acceder a ella o directamente bloquea el acceso.
- **Auditoría de apps**: Analiza y muestra los derechos de acceso y propósito de las apps instaladas, identifica posibles riesgos que afecten a la privacidad de los datos, para que el usuario sepa realmente cuánta información está proporcionando a cada app.
- **Bloqueo de SMS/llamadas**: Filtra remitentes de spam, números incorrectos y mensajes SMS y MMS no deseados.
- **Control parental**: Proteger a los niños contra amenazas al navegar por internet o intentar usar/instalar aplicaciones no seguras.
- **Copia de seguridad**: Permiten hacer copias de seguridad de los contactos, sms.

2.7.3. LISTADO DE HERRAMIENTAS

La Tabla 2.4 con 16 herramientas, que si bien no son ni mucho menos todas la herramientas que existen hoy en día en el mercado, sí que son las que hemos considerado que son más conocidas, tienen mejores puntuaciones y funcionalidades más amplias de cuantas hemos revisado en diferentes sitios web que analizan soluciones de antivirus.

EMPRESA	PRODUCTO	CARACTERÍSTICAS	GRATUITO
Avast Software	Mobile Security & Antivirus (https://www.avast.com/es-es/free-mobile-security)	Antivirus; Gestor de aplicaciones; Auditoría de aplicaciones; Navegación segura; Cortafuegos; Protección antirrobo; Filtrados de SMS/llamadas.	Si (Tiene opciones Premium de pago)
Avira	Avira Antivirus Security (https://www.avira.com/es/free-antivirus-android)	Antivirus; Protección de privacidad; Protección antirrobo; Protección de identidad; Administración del dispositivo.	Si (Tiene opciones Premium de pago)
AVG	AVG Antivirus (http://www.avg.com/es-es/antivirus-for-android)	Antivirus; Protección en línea, Protección antirrobo; Bloqueador de llamadas; Copia de respaldo, Bloqueo de aplicaciones; Bloqueo de dispositivo; Bloqueo de publicidad.	Si (Tiene opciones PRO de pago)
Bitdefender	Bitdefender Mobile Security (http://www.bitdefender.es/solutions/mobile-security-android.html)	Antimalware; Protección antirrobo; Navegación segura; Bloqueo de apps.	No
eScan	eScan - Mobile Antivirus (http://www.escanav.com/en/android-antivirus/mobile-security-for-android.asp)	Antivirus; Antirrobo; Filtrado de llamadas y SMS; Control parental; Control de aplicaciones; Navegación segura; Chequeo de las apps intaladas.	No

ESET	ESET NOD32 Mobile Security para Android (http://www.eset.es/particulares/movil/)	Antivirus; Protección de navegación; Protección antirrobo; Filtro de llamadas y mensajes; Control del dispositivo; Auditoría de aplicaciones.	Si (Tiene opciones Premium de pago)
F-Secure	F-Secure Mobile Security (https://www.f-secure.com/en/web/home_global/mobile-security)	Antivirus y antispyware; Protección antirrobo; Escaneo de aplicaciones; Protección de navegación; Bloqueo llamadas y SMS; Control parental.	No
Ikarus	IKARUS mobile.security (http://www.ikarussecurity.com/es/soluciones/todas-las-soluciones/endpoint-protection/ikarus-mobilesecurity/)	Antivirus; Navegación segura; Filtro URL; Protección antirrobo; Protección de tarjeta SIM; Filtrado de SMS o correos electrónicos.	Si (Tiene opciones Premium de pago)
Iobit	AMC Security (http://mobile.iobit.com/)	Antimalware; Protección antirrobo; Navegación segura; Auditoria de apps; Antispam; Copia de seguridad.	Si (Tiene opciones Premium de pago)
Intel Security	McAfee Mobile Security (http://home.mcafee.com/store/mobile-security)	Antivirus; Protección de privacidad (perfiles de aplicaciones, alertas sobre aplicaciones, bloqueo de aplicaciones, filtro de llamadas y SMS); Optimización rendimiento del dispositivo; Protección antirrobo; Protección web; seguridad en Wi-Fi.	No
Kaspersky Lab	Kaspersky Internet Security for Android (http://www.kaspersky.es/software-antivirus-domestico/android-security)	Protección antimalware; Protección web; Protección antirrobo; Protección de la privacidad; Filtro de llamadas y mensajes.	No

Symantec	Norton Mobile Security (https://es.norton.com/norton-mobile-security)	Protección antimalware; Protección antirrobo; Protección de la privacidad; Filtro de llamadas y mensajes; Navegación segura.	Si (Tiene opciones Premium de pago)
Sophos	Sophos Mobile Security for Android (https://www.sophos.com/en-us/products/free-tools/sophos-mobile-security-free-edition.aspx)	Protección antivirus y antimalware; Protección antirrobo y contra pérdidas; Protección contra Spam; Protección de la privacidad; Navegación segura.	Si
Trend Micro	Mobile Security & Antivirus (http://www.trendmicro.es/productos/mobile-security-for-android/index.html)	Protección frente a virus; Protección frente a robo de datos; Navegación segura; Filtrado de llamadas y texto; Función antirrobo; Privacidad en redes sociales; Optimización del sistema.	Si (Tiene opciones Premium de pago)
TrustPort	Mobile Security (http://www.trustport.com/en/home-users)	Antivirus; Gestor de aplicaciones; Copia de contactos; Antispam; Antirrobo; Control parental; Navegación segura.	Si (Tiene opciones Premium de pago)
WEBROOT	SecureAnywhere Mobile (http://www.webroot.com/us/en/home/products/android)	Antivirus; Navegación segura; Protección antirrobo; Bloqueo de llamadas y SMS; Borrado remoto des dispositivo; Inspector de apps; Protección contra desinstalación.	Si (Tiene opciones Premier de pago)

Tabla 2.4 – Herramientas de protección para Android

3. OBJETIVO Y METODOLOGÍAS DE EVALUACIÓN

En este capítulo definimos el objetivo del TFM, ya que teniendo en cuenta que se trata de un trabajo de tipo experimental hemos de realizar pruebas, así pues, en el siguiente apartado explicaremos qué se pretende conseguir con este TFM.

En los apartados posteriores del capítulo precisaremos la metodología que vamos a seguir para evaluar las herramientas de prevención existentes. Hay dos metodologías debido a que, en la primera, el entorno donde se desarrollan las pruebas es virtual, particularidad que impide medir cómo afecta a la autonomía del dispositivo el uso de las herramientas de prevención y, en la segunda las pruebas se desarrollan sobre un dispositivo físico. De esta manera ambas metodologías precisan utilizar entornos y herramientas de monitorización diferentes.

3.1. OBJETIVO

El objetivo del TFM es evaluar un conjunto de herramientas de prevención existentes en el mercado y determinar si son eficaces contra el malware, además comprobaremos cómo afecta al rendimiento del dispositivo en cuanto a consumo de CPU, memoria y durabilidad de la batería.

Para realizar la evaluación, en primer lugar haremos una selección del malware, lo que nos requerirá tener acceso a muestras para poder escoger las que nos interesen según los criterios que hemos decidido y que detallaremos más adelante. A continuación habremos de seleccionar las herramientas de prevención a evaluar de entre la multitud de herramientas que existen hoy en día en el mercado (como hemos puesto de relieve en el punto 2.7.3) y seguidamente aplicaremos las metodologías que explicamos a continuación para obtener los resultados de la evaluación.

En base al estudio realizado de la situación actual (Capítulo 2) y a los resultados obtenidos de la evaluación que realizaremos, el TFM finalizará con una guía de buenas prácticas para que los usuarios de Android puedan estar lo más seguros posible contra el malware.

3.2. METODOLOGÍA EFECTIVIDAD HERRAMIENTAS Y RENDIMIENTO

La metodología que utilizaremos para evaluar la efectividad de las herramientas y el impacto que tiene sobre el dispositivo en cuanto a rendimiento será la siguiente:

1. Crear un entorno virtual, concretamente una plantilla (la versión Kit-Kat 4.4.4[58]), con Android en el pc. Lo hemos realizado con Android-x86[59]. Y el entorno virtual que hemos utilizado es el VirtualBox v.5.0.10
2. Instalar la utilidad necesaria para analizar el rendimiento. Hemos considerado que la herramienta 3C ToolBox Pro[60] nos ofrecía lo que requerimos.
3. Instalar la app que nos permitirá transferir ficheros entre el PC y el sistema virtual (esta operación la realizaremos con Airdroid[61].).
4. Realizar una prueba de análisis de rendimiento con el sistema “limpio”. Que posteriormente servirá de referencia.
5. Crear una máquina virtual desde esa plantilla. Haremos un clon.
6. Instalar una herramienta de prevención a testear y actualizarla a la última versión.
7. Arrancar la grabación del entorno de ejecución con la herramienta de análisis de rendimiento, teniendo el antivirus operativo.
8. Pasar el conjunto de malware al dispositivo, es decir, transferir los ficheros entre el pc y el dispositivo virtual.
9. Determinar si la herramienta de protección los detecta en tiempo real (cuando se copien al sistema de ficheros), o por el contrario hemos de forzar un análisis del sistema y comprobar cuántos detecta del conjunto.
10. Extraer los resultados de la monitorización del sistema en cuanto al rendimiento.
11. Eliminar la máquina virtual y volver al punto 5.

3.3. METODOLOGÍA ANÁLISIS CONSUMO DE BATERÍA

Teniendo en cuenta que el entorno donde vamos a realizar las pruebas de efectividad de las herramientas de prevención y de rendimiento es virtual, no podemos realizar un análisis para comprobar en qué medida afectan estas herramientas a la durabilidad de

la batería. Así pues, vamos a utilizar un dispositivo físico para realizar cómo afectan estas herramientas en la reducción de autonomía. Este dispositivo físico será un Samsung Galaxy S5 (concretamente el modelo SM-G901F) con la versión de Android 5.0.2 (Lollipop).

La metodología a seguir para obtener los resultados será la siguiente:

1. Tener el S.O. Android recién instalado.
2. Instalar la herramienta de análisis de batería. GSAM battery monitor PRO[62]. Nos mostrará, desde la última carga en qué se ha usado la batería.
3. Cargar la batería al 100%
4. Instalar herramienta prevención.
5. Generar actividad durante un tiempo (transferir ficheros, navegar, gestionar correo, etc.) simulando un funcionamiento habitual hasta que la batería se descargue hasta el 50% como mínimo, para que la herramienta de monitorización obtenga datos.
6. Recopilar resultados de GSAM battery monitor PRO.
7. Desinstalar herramienta prevención.
8. Si no es la última herramienta a evaluar. Volver al paso 3

4. MALWARE A ESTUDIAR

Este capítulo muestra la selección del software malicioso a testear, en base a las tipologías de malware más común y a las familias existentes. Además, cada una de las muestras seleccionadas vamos a explicarlas en cuanto a cuál es su firma, qué hacen, qué tipo de fichero, tamaño de la muestra, etc.

4.1. TIPOLOGÍAS DE MALWARE MÁS COMÚN

Si analizamos diferentes informes anuales sobre amenazas y/o malware en dispositivos móviles de compañías de seguridad informática vemos que:

- En el año 2012, según la compañía TrendMicro[63], el malware relacionado con el envío de SMS a servicios Premium fue el más difundido, concretamente la familia OpFake. Seguidamente fue la familia de malware FakeInst, y entre las dos ya copaban más del 55% del malware. A continuación mostramos la figura 4.1 con el

Ranking de las 10 familias de malware que detectó la compañía.

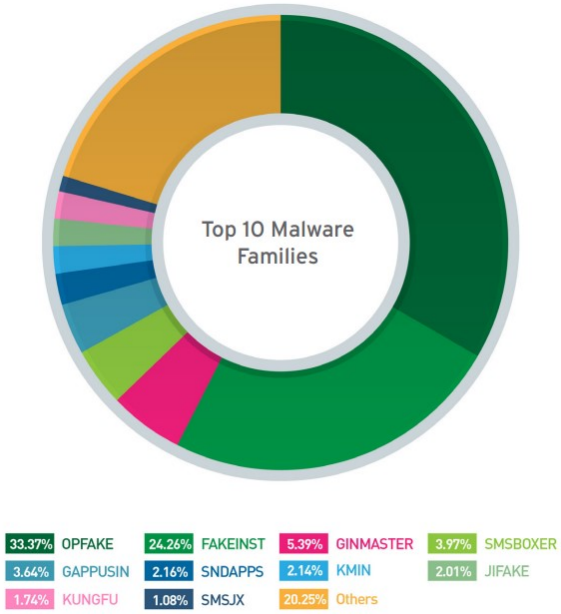


Figura 4.1 - Ranking familias Malware año 2012. Fuente [i]

También en el año 2012, pero centrándonos en el último trimestre, vemos que la compañía F-Secure[64] detectó 96 nuevas familias y variantes de amenazas para Android, de las cuales, un porcentaje elevado tiene que ver con el envío de SMS a servicios Premium. Como familia principal que han detectado aparece Airpush (con un 13,5%), seguidamente está AdWo (con un 11,8%), a continuación algo que categorizan de forma genérica Heuristic-Malware (con un 10,4%), i sigue Boxer.C (con un 9,9). Las siguientes categorías ya representan menos de un 10% del total.

- En el año 2013[65][66][67], según F-Secure, las familias de malware más representativas detectadas por sus sistemas fueron: En el primer semestre Badnews, FakeKrbank, VMVol, FakeDefender, Pincer y Obad en el primer semestre y Fakeinst, GinMaster, OpFake, SmsSend, QDPluguin, InfoEtealer y Vdloader en el segundo.
- En el año 2014 según PulseSecure[68] la capacidad de obtener beneficio de un usuario final con servicios SMS Premium o redes de anuncios (ad networks) está presente en todas y cada una de las 10 familias más detectadas durante el año. Éstas son: Smsreg, Agent, Dowgin, Stealer, Fakeinst, SMSager, Andr, Waps Airpush y Kuguo.

Como conclusión, del periodo analizado, podemos decir que el principal objetivo de los desarrolladores de malware es el beneficio económico y de una manera rápida y sencilla. Para conseguir estos objetivos, en todos los años se aprecia que el envío de

mensajes SMS a servicios Premium y/o las redes de publicidad son los malware más populares y más extendidos.

4.2. SELECCIÓN DE MALWARE

Las muestras de malware seleccionadas las hemos obtenido de www.virusshare.com que es un repositorio de software malicioso disponible para la comunidad de investigadores de seguridad informática, analistas forenses, etc. El acceso al repositorio solamente es posible vía invitación del administrador del sitio web, en respuesta a una petición previa explicando quien pide autorización para acceder y los motivos por los cuales se requiere el acceso a las muestras de malware.

Para la selección que hemos realizado, como ya hemos comentado anteriormente, nos hemos basado en la información que hemos generado del **Anexo I**, donde se dispone de una tabla con las familias de malware actualmente existentes, y las tipologías de malware más común. Con esta base hemos obtenido muestras del repositorio que consideramos son representativas de un amplio espectro de actividades que el malware puede realizar.

La Tabla 4.1 expone las muestras elegidas con información que consideramos relevante de las mismas. En caso que se desee más información de cada una de las muestras en cuanto a permisos requeridos para que el malware tenga éxito, las actividades del apk, servicios, receptores, strings que contiene, etc. Es decir, información técnica de detalle, se puede ir a la web de virustotal[69] e introducir la firma MD5 de la muestra (identifica unívocamente la muestra) para realizar la búsqueda.

FAMILIA	FIRMA MD5 MUESTRA	TAMAÑO	TIPO DE FICHERO	FECHA APARICIÓN
Ackposts	63ccc6662c83e83cbe0432b4a0bc4058	4.919.787 bytes	ZIP	27/03/2014
Este troyano roba información de contactos desde el dispositivo comprometido y los carga en un servidor remoto.				
Airpush/StopSMS	644323ef97e7fcea7710ee4fa17e28c7	3.060.303 bytes	ZIP	18/03/2014
Airpush es una, muy agresiva, red de anuncios (Ad-Network).				
Arspam	e7584031896cb9485d487c355ba5e545	115.856 bytes	ZIP	04/06/2013
Este malware representa la primera etapa de la piratería por motivos políticos (hacktivismo) en plataformas móviles.				
Basebridge	c3b9ed157b71fba7c01be4394c12cd01	1,441,300 bytes	ZIP	03/05/2012
Envía datos confidenciales (SMS, IMSI, IMEI) a un servidor remoto.				
Dowgin	75a7dd9650d153e021ce4d1b419cb470	538,197 bytes	ZIP	30/11/2013
Muestra publicidad y envía información de las apps instaladas, IMEI, etc. a un servidor remoto. Descarga e instala nuevas apps.				
DroidDream	aa1f2dcdecba29a55050809aee030077	14.079 bytes	ZIP	04/07/2012
Utiliza dos herramientas diferentes (rageagainstthecage y exploid) para rootear el dispositivo.				
DroidKungfu	f438ed38b59f772e03eb2cab97fc7685	574.864 bytes	ZIP	17/05/2013
Recoge información en el teléfono infectado, que se vuelca en un archivo local y después se envía a un servidor remoto.				
FakeDefender	5290df867914473426b82233567c03af	2.735.848 bytes	ZIP	27/11/2014
Muestra falsas alertas de seguridad y engaña para comprar una aplicación con el fin de eliminar el malware inexistente en el dispositivo.				
FakeFlash	1369b2afbd3d80856cbed9fecb5e4e51	400.939 bytes	ZIP	15/12/2014
Este troyano redirige al usuario a través de proxies de pago.				
FakeInst	41ca3efde1fb6228a3ea13db67bd0722	65.207 bytes	ZIP	23/11/2014
Estas aplicaciones envían mensajes SMS a servicios premium.				
FakeTimer	c4d631d2ded1f20bcd752d573be707da	79.728 bytes	ZIP	30/03/2012
Envía la información personal a un servidor remoto y abre páginas web pornográficas.				
Foncy	1a3fb120e5a4bd51cb999a43e2d06d88	16.927 bytes	ZIP	29/07/2012

Envía mensajes SMS a servicios premium.				
Geinimi	e0106a0f1e687834ad3c91e599ace1be	570.420 bytes	ZIP	07/07/2013
Abre una puerta trasera y transmite información desde el dispositivo (IMEI, IMSI, etc.) a un URL específico.				
HongTouTou/Adrd	5895bcd066abf6100a37a25c0c1290a5	276.333 bytes	ZIP	04/07/2012
Roba información que posteriormente carga, a través de un proxy local, a un servidor remoto. Los datos se cifran de antes de enviarlos.				
Jifake	37a46aec9aa86831faa3ddb6b05a05f8	1.256.548 bytes	ZIP	04/07/2012
Esta aplicación envía mensajes SMS a servicios premium.				
Ksapp	818ef792fb4e20e0fe89feb35b1709b1	6.166.888 bytes	ZIP	29/07/2013
Maneja conexiones de acceso remoto, ataques de DoS o DDoS, captura de teclado, elimina archivos u objetos, o termina procesos.				
Locker/SLocker Ransomware	645a60e6f4393e4b7e2ae16758dd3a11	488.296 bytes	ZIP	26/11/2014
Este troyano es el primer CriptoLocker para Android.				
Obad	f7be25e4f19a3a82d2e206de8ac979c8	85.079 bytes	ZIP	17/11/2014
Envía SMS a servicios premium, baja otros programas de malware y/o los envía a través de Bluetooth. Ejecuta remotamente comandos.				
OpFake	20c593c08fca752a7cfd10e9b3af566b	35.790 bytes	ZIP	21/02/2013
Estas aplicaciones envían mensajes SMS premium.				
RootSmart/Bmaster	f70664bb0d45665e79ba9113c5e4d0f4	314.445 bytes	ZIP	07/11/2012
Aprovecha la vulnerabilidad GingerBreak para obtener privilegios de root.				
SMSreg	94aee9e957c0ffc7686a068b660f01b9	1.552.196 bytes	RAR	08/04/2014
Registra el dispositivo infectado a servicios de pago.				
Spitmo	cfa9edb8c9648ae2757a85e6066f6515	19.724 bytes	ZIP	22/06/2012
Roba información del dispositivo. También intercepta mensajes SMS de los bancos (mensajes mTAN) y los carga a un servidor remoto.				
TapSnake/Droisnake	9756e14eb74ba3f5a339f749987b6a47	25.968 bytes	Dalvik dex	04/05/2012
Publica la ubicación del teléfono en un servicio web.				
Tesbo	66ae7dc1cf84b12484db7b21b9ef960f	397.026 bytes	ZIP	09/12/2013
Envía información del dispositivo a servidores remotos.				

Titan	490a20ecaa5ddba94fb33fd9139a9857	502.288 bytes	JAR	10/04/2015
Ataques dirigidos en Asia y trata de filtrar información sensible. Se propaga a través de mensajes SMS.				
UpdtKiller	ad377c9a4e579e426c0f22b968e83401	4.153.617 bytes	ZIP	24/09/2013
Detecta y desactiva aplicaciones AV instalados.				
USBcleaver	c22c068eaee7ad7fd4fd015cd50045db	57.440 bytes	ZIP	25/11/2013
Cuando el dispositivo está conectado a un Windows con ejecución automática activada, obtiene información de la computadora.				
Waps/Simhosy	d23995ef588eb295039ae96014fa21eb	593.282 bytes	ZIP	01/01/2014
Intenta robar mensajes SMS y las entradas de contactos en el dispositivo infectado.				
ZertSecurity	1cf41bdc0fdd409774eb755031a6f49d	100.878 bytes	ZIP	12/05/2013
Intenta engañar al usuario para que inserte sus detalles de cuenta bancarios que se enviarán a los atacantes.				
Zitmo/Citmo	ecbbce17053d6eaf9bf9cb7c71d0af8d	19.865 bytes	ZIP	17/04/2012
Trata de robar códigos confidenciales bancarios de autenticación (mensajes mTAN) enviados al dispositivo infectado.				

Tabla 4.1 – Malware seleccionado para el testeo de las herramientas de prevención

5. SELECCIÓN HERRAMIENTAS DE PREVENCIÓN A EVALUAR

De las herramientas listadas en el apartado 2.7.3 hemos escogido diez para hacer la evaluación. La selección la hemos realizado en base a las comparativas realizadas por dos organizaciones independientes[70] que se dedican a realizar test sistemáticos de software de seguridad. Hemos elegido varias de las mejores puntuadas[71][72], también alguna de las que no han obtenido tan buenos resultados y algunas que tienen un nombre comercial muy importante dentro del sector de la seguridad informática.

Para empezar la lista escogemos BitDefender ya que en ambas listas ha conseguido el mayor número de aciertos. Seguimos con ESET ya que en av-test ha obtenido el segundo lugar y continuamos con Trend Micro que en av-comparatives también está en segunda posición. Otras soluciones que también están en ambas listas y que también han obtenido muy buenos resultados son: Sophos, Avira, Avast, Kaspersky, McAfee y para finalizar nuestra selección escogemos Ikarus y AVG.

De esta forma los productos que vamos a evaluar son:

1. AVG antivirus de la empresa AVG.
2. Avira Antivirus Security de la empresa Avira.
3. Bitdefender Mobile Security de la empresa Bitdefender.
4. ESET NOD32 Mobile Security para Android de la empresa ESET.
5. Ikarus mobile.security de la empresa IKARUS.
6. Kaspersky Internet Security for Android de la empresa Kaspersky Lab.
7. McAfee Mobile Security de la empresa Intel Security.
8. Mobile Security & Antivirus de la empresa Trend Micro.
9. Mobile Security & Antivirus de la empresa Avast Software.
10. Sophos Mobile Security for Android de la empresa Sophos.

Figura 5.1 – Logos de las empresas de herramientas de prevención de evaluar.

6. RESULTADOS EVALUACIÓN

En este capítulo vamos a presentar los resultados de la evaluación, de las diez herramientas de prevención seleccionadas, respecto a la efectividad en cuanto a la detección de las treinta muestras de malware. También mostraremos qué impacto en el rendimiento del sistema conlleva tener instalada una herramienta de este tipo. Este impacto lo evaluaremos en tres aspectos que consideramos clave para el buen funcionamiento de un dispositivo móvil, consumo de CPU, de memoria RAM y de batería.

6.1. EFECTIVIDAD HERRAMIENTAS

En el **Anexo II** presentamos la matriz con los resultados detallados obtenidos de nuestra evaluación. Las diez columnas se corresponden con las herramientas a testear y las treinta filas con las muestras de malware. De esta forma, en cada cruce entre fila y columna consta el nombre de cómo cada herramienta de prevención ha identificado la muestra de malware seleccionada y, en caso que no se haya detectado el fichero como malware, se ha rellenado el campo de color rojo. Si hacemos una primera aproximación a los mismos, nos muestran varias evidencias:

- Todas las muestras han sido reconocidas por alguna de las soluciones. Este hecho nos indica que todas las muestras de la selección realizada en el punto 4.2 son válidas. Además, como prueba adicional y para evitar falsos positivos, las muestras que sólo las han reconocido alguna herramienta de las escogidas, las hemos buscado en virustotal y, en todos los casos, han aparecido multitud de otras herramientas que sí que las reconocen como software malicioso.
- Ninguna solución ha reconocido el 100% de las muestras como malware. Ello nos lleva a concluir que no existe la seguridad total.
- El resultado es independiente de si las herramientas son de pago o gratuitas. Es decir, a priori se podría pensar que las soluciones de pago, por el hecho de serlo, obtendrían los mejores resultados, pues esto no ha sido así.

Si analizamos con más detalle los resultados, vemos que la clasificación queda como sigue:

- La mejor solución de prevención de malware para la muestra que hemos planteado ha sido Ikarus mobile.security de la empresa IKARUS con un acierto en 28 muestras sobre 30. Es decir, una detección del 93,33%.
- Seguidamente ha habido un triple empate, con una detección de 27 muestras sobre las 30 planteadas, entre las herramientas de AVG, Avira y ESET.
- Les sigue SOPHOS, con 26 aciertos sobre 30, ha obtenido un 86,67% de aciertos.
- Tenemos un empate entre BitDefender y Kaspersky con una detección del 83,33% de las muestras. Han obtenido 25 aciertos sobre 30 muestras.
- Seguimos con otro empate entre Intel Security y Trend Micro habiendo detectado cada una 24 muestras de las 30 posibles, o lo que es lo mismo un 80%
- Finaliza la clasificación AVAST con una detección del 73,33%. Ha detectado 22 de las 30 muestras.

Adjuntamos gráfica, en la Figura 6.1, donde se muestran de una forma visual los resultados obtenidos en cuanto a la efectividad de las herramientas. Vemos que están ordenadas de izquierda a derecha en cuanto al porcentaje de detección.

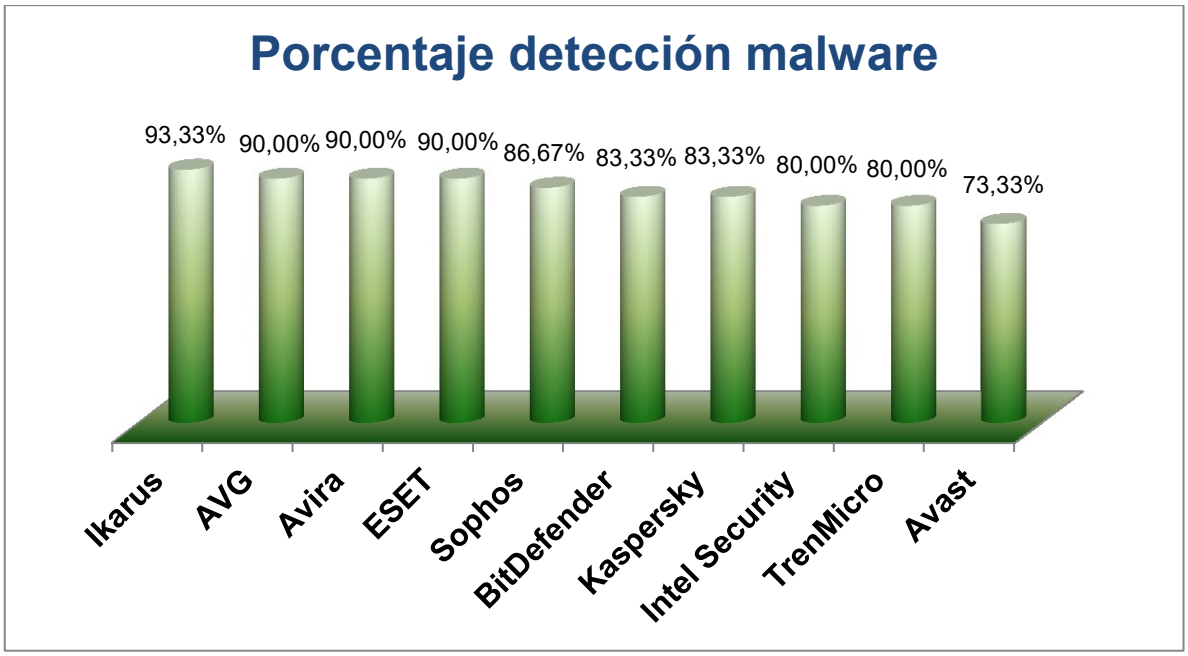


Figura 6.1 – Gráfica resultados porcentaje detección de malware.

Otras consideraciones sobre las herramientas, que nos ofrecen los resultados, son que cada empresa pone los nombres que considera oportunos a las muestras detectadas, así pues, hay muchas muestras que las detectan varias soluciones y están identificadas como si fueran de familias diferentes de malware. Por ejemplo, la muestra de la familia Titan tiene varias denominaciones, en cuanto a la familia, como pueden ser Titan, Tediss y Raidum. Además, vemos que AVG ha detectado varias muestras como malware pero no les ha puesto nombre y otras muchas de ellas las identifica como Android/deng, de forma genérica. En cuanto a la herramienta de Kaspersky también vemos que la mayoría de las muestras detectadas las identifica como UDS:DangerousObject.Multi.Generic sin especificar a qué familia pertenecen ni que variante son.

También vemos que, en la identificación que hacen muchas de las empresas del malware detectado, en el mismo nombre indican de forma implícita:

- El sistema operativo dónde es operativo el malware. Ejemplos: Android:, Android/, AndroidOS, etc.
- Qué tipo de malware es. Si se trata de un troyano (Trojan, [Trj], TrojanSMS, etc.), si es spyware, Adware, PUA, etc.
- La familia a la que pertenece el malware (Obad, Genimi, Zitmo, etc.)
- La variante que es (A, C, B, ACE, etc.)

En otro aspecto del análisis de los resultados, y centrándonos en el malware, vemos que

hay una muestra de malware, perteneciente a la familia SMSReg que solamente la ha detectado la herramienta de Ikarus y, hay otras dos muestras, que son FakeInst y Foncy que sólo las han detectado dos herramientas. En el primer caso Avira e Ikarus y en el segundo Btidefender y Sophos. Teniendo en cuenta que se tiene conocimiento de este malware desde hace tiempo (SMSReg y FakeInst es del año 2014; Foncy es del año 2012) es cuanto menos preocupante ver cómo es posible que tan pocas soluciones de seguridad las hayan detectado. Algo más tranquilizador es el hecho que observamos como Ackposts, Arspam, BaseBridge, FakeFlash, FakeTimer y unas cuantas muestras más, de familias diferentes, han sido detectadas por todas las herramientas evaluadas.

Otro detalle a considerar, no tanto de los resultados obtenidos, si no en cuanto a la forma de detectar el malware, es que solamente dos herramientas, Avast e Ikarus, han sido capaces de descubrir en tiempo real el malware que se estaba copiando en el sistema Android de test. Todos los otros fabricantes, cuando transferíamos el malware desde el pc al sistema virtual Android (con la herramienta Airdroid) han sido incapaces de detectarlo y para conseguirlo hemos tenido que forzar un análisis de forma manual. Ha habido incluso, la herramienta Bitdefender, que no detectaba el malware copiado en su sistema de ficheros ni forzando el análisis manual y, hemos tenido que renombrar los archivos con el malware para agregarles la extensión .apk y forzar de nuevo el escaneo para que los detectara.

6.2. IMPACTO EN EL RENDIMIENTO DEL SISTEMA

Tal como ya hemos mencionado con anterioridad, el impacto en el rendimiento del sistema lo vamos a medir en tres aspectos que consideramos fundamentales. Éstos son, el consumo de CPU, la memoria RAM utilizada y el consumo de batería. En primer lugar vamos a presentar los resultados del consumo de CPU y memoria, ya que esta información la hemos extraído del entorno virtualizado. A continuación, mostraremos los resultados en el consumo de batería, medidas que hemos tenido que tomar en el dispositivo físico.

6.2.1. CONSUMO DE CPU Y MEMORIA RAM

En general, podemos afirmar que hemos detectado un consumo bastante pequeño, en cuanto a estas variables, de las soluciones evaluadas. Sí que es verdad que cuando se han forzado análisis de forma manual ha habido picos de utilización de CPU (en algunos casos elevados) y de memoria, pero nada que el sistema no pudiera asumir.

A continuación vamos a ir mostrando los resultados de cada solución de forma

individualizada, pero antes que nada también mostramos, en la Figura 6.2, lo que hemos recopilado sin que haya ninguna solución de antivirus instalada.

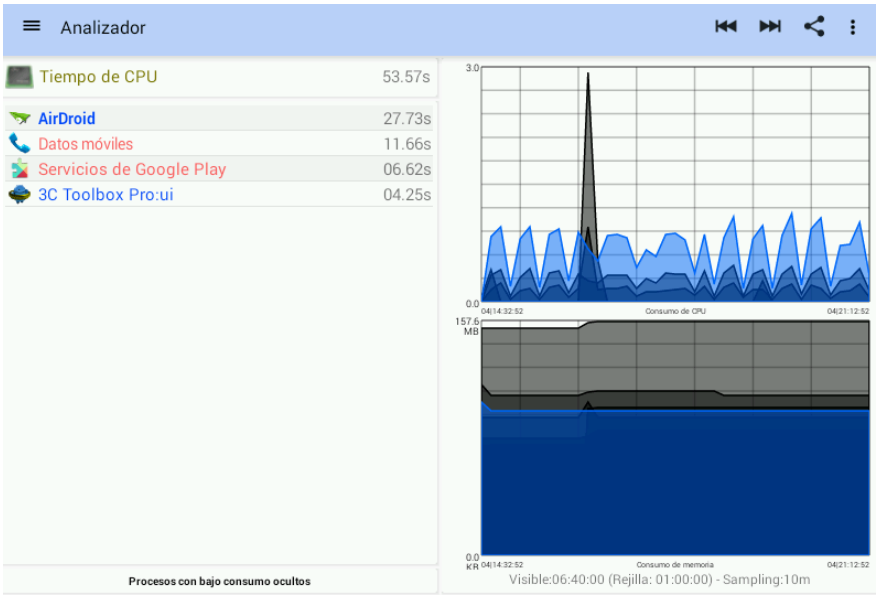


Figura 6.2 – Consumo de CPU y memoria sin herramienta de prevención instalada.

Para cada solución instalada y evaluada presentamos:

- Dos gráficas del tiempo/uso de CPU y de consumo de memoria del proceso de la herramienta de protección. La primera (a la izquierda) muestra una lista de todos los procesos en ejecución que el sistema tenía en ese momento y una gráfica con la información superpuesta de todos ellos. La segunda (a la derecha) muestra los mismos datos pero sólo mostrando la gráfica del proceso en cuestión.
- Una imagen con los detalles del proceso, información relativa al Id del proceso, cuando se inició, prioridad dentro del sistema, etc. Lo que nos concierne, y nos hemos de centrar en este caso, está en la parte derecha que muestra el Uso de CPU y memoria.
- Un pequeño resumen mencionando los valores obtenidos con la herramienta de medición de rendimiento.

AVAST

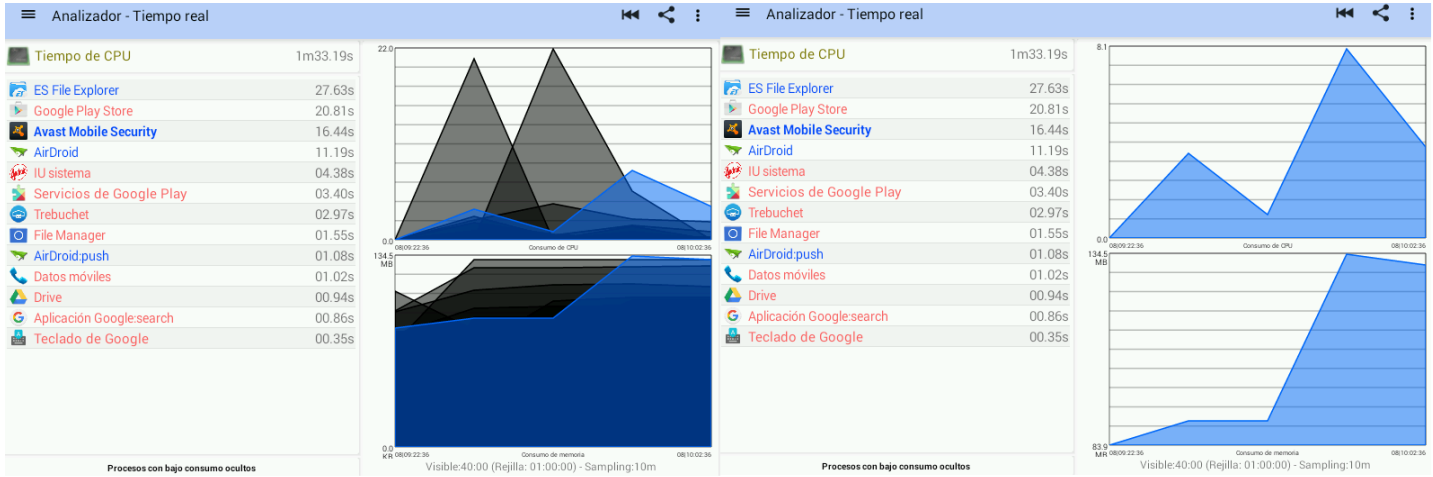


Figura 6.3 – Consumo de CPU y memoria herramienta de Avast

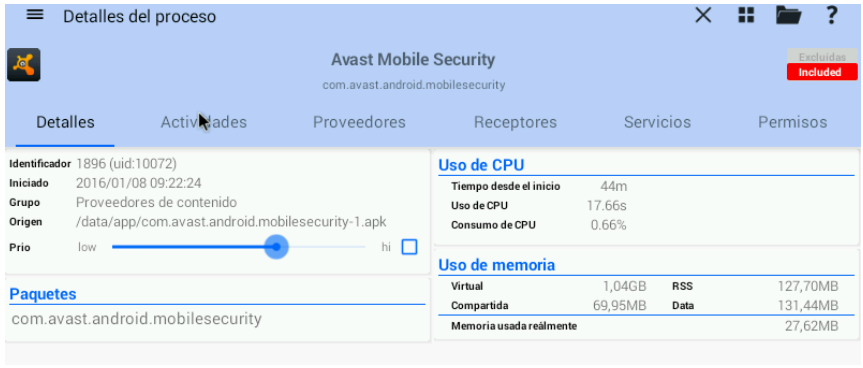


Figura 6.4 – Detalle del proceso Avast Mobile Security

Los resultados muestran que el consumo máximo de CPU del proceso ha sido de un 8,1% y la necesidad de memoria ha sido de 134,5 MB (el sistema tiene 1GB). Además, en la información que nos ofrece el detalle del proceso vemos que sólo ha usado un 0,66% de CPU desde que está iniciado el proceso y la memoria usada realmente es de 27,62MB.

AVG

Como podemos observar en las gráficas, el consumo máximo de CPU del proceso ha sido de un 22,5% y en cuanto a la memoria ha necesitado 137,9 MB. Además, en la información que nos ofrece el detalle del proceso vemos que sólo ha usado un 1,06% de CPU desde que está iniciado el proceso y la memoria usada realmente es de 26,65MB.

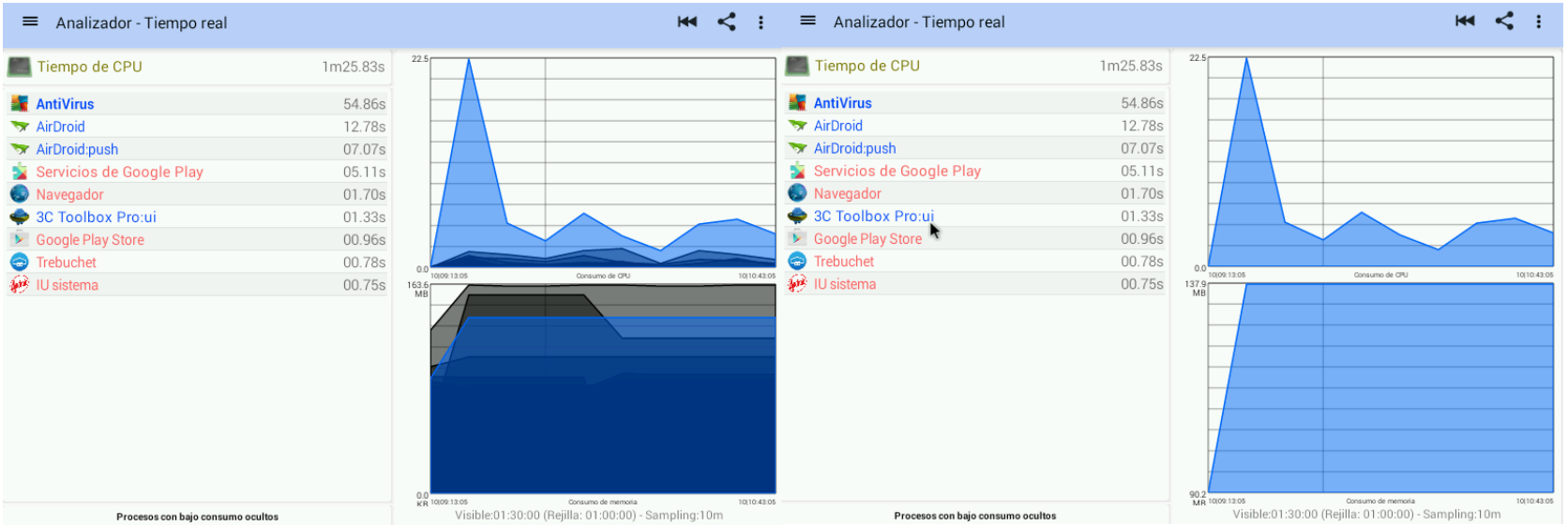


Figura 6.5 – Consumo de CPU y memoria herramienta de AVG

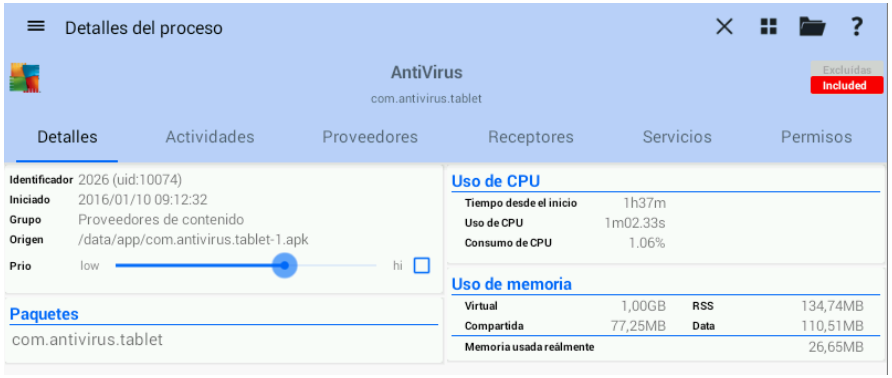


Figura 6.6 – Detalle del proceso AVG

AVIRA

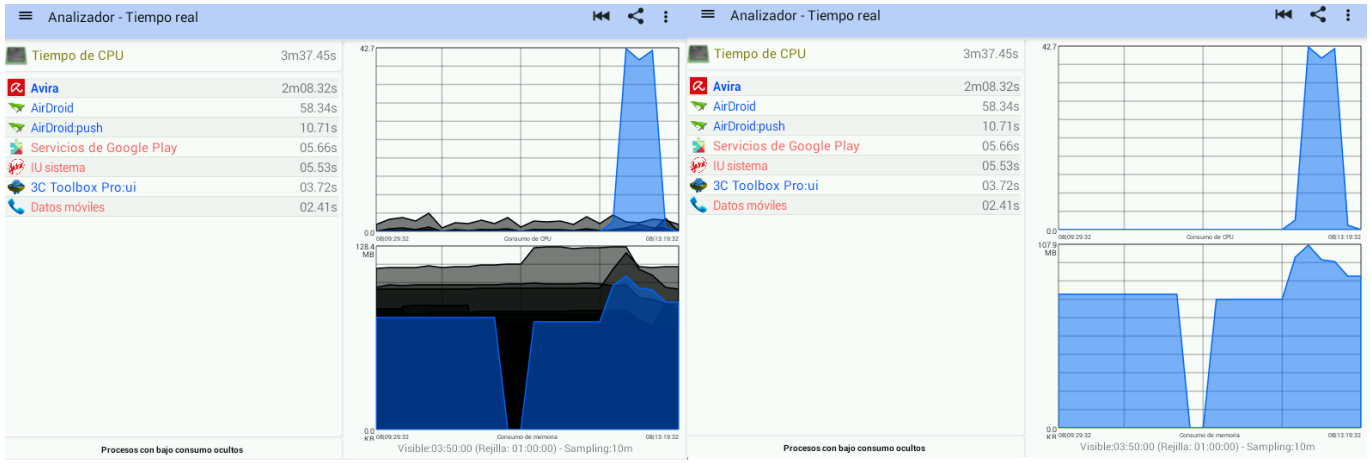


Figura 6.7 – Consumo de CPU y memoria herramienta de Avira

Figura 6.8 – Detalle del proceso Avira

Si analizamos los resultados obtenidos, vemos que el consumo máximo de CPU del proceso ha sido de un 42,7% y la necesidad de memoria ha sido de 107,9 MB. Además, en la información que nos ofrece el detalle del proceso vemos que sólo ha usado un 2,06% de CPU desde que está iniciado el proceso y la memoria usada realmente es de 28,95MB.

BITDEFENDER

Lo primero que sorprende es que las gráficas están en formato vertical en lugar de horizontal como las otras. Esto es debido a que no pudimos hacer funcionar la solución en la plataforma virtual y la tuvimos que testear en el dispositivo físico. Igualmente observamos que el consumo máximo de CPU del proceso ha sido de un 58,7% y la necesidad de memoria ha sido de 125,8 MB. Además, en la información que nos ofrece el detalle del proceso vemos que sólo ha usado un 1,28% de CPU desde que está iniciado el proceso y la memoria usada realmente es de 5,93MB.

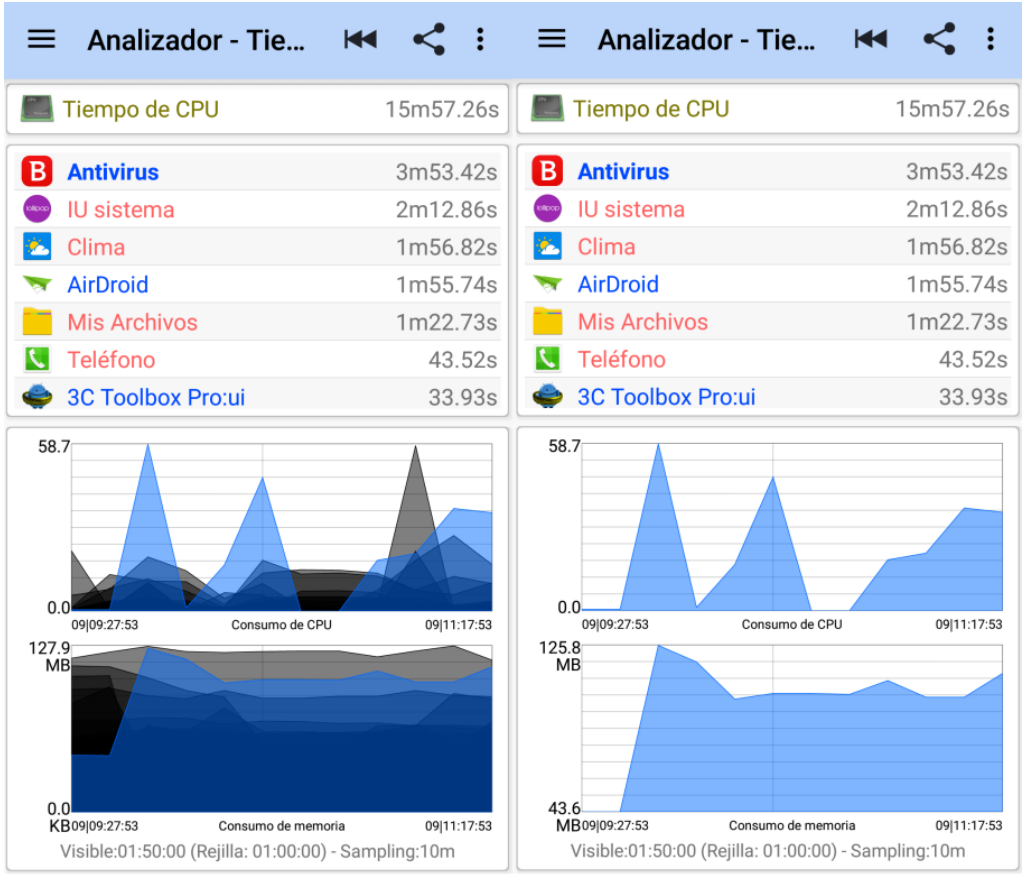


Figura 6.9 – Consumo de CPU y memoria herramienta de Bitdefender

Figura 6.10 – Detalle del proceso Bitdefender

ESET

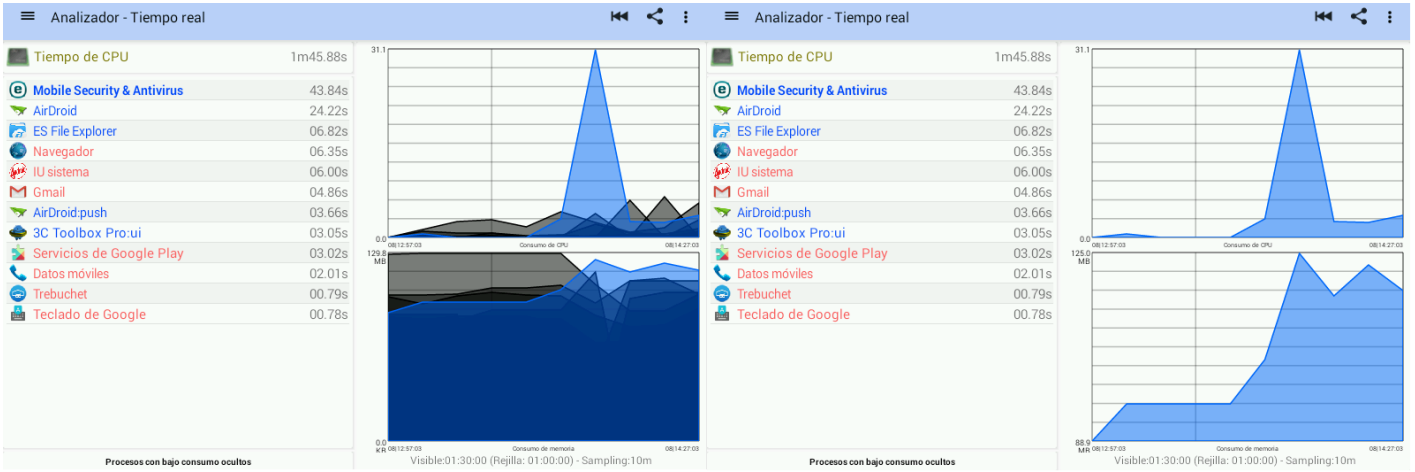


Figura 6.11 – Consumo de CPU y memoria herramienta de ESET

Figura 6.12 – Detalle del proceso ESET

A la luz de los resultados obtenidos, concluimos que el consumo máximo de CPU del proceso ha sido de un 31,1% y la necesidad de memoria ha sido de 125 MB. Además, en la información que nos ofrece el detalle del proceso vemos que sólo ha usado un 0,80% de CPU desde que está iniciado el proceso y la memoria usada realmente es de 28,25MB.

IKARUS

En las gráficas, vemos que el consumo máximo de CPU del proceso ha sido de un 93,9%, el valor más alto de toda la serie de mediciones, cuando hemos ejecutado un análisis bajo demanda y la necesidad de memoria ha sido de 130,6 MB. Además, en la información que nos ofrece el detalle del proceso vemos que sólo ha usado un 2,24% de CPU desde que está iniciado el proceso y la memoria usada realmente es de 81,44MB.

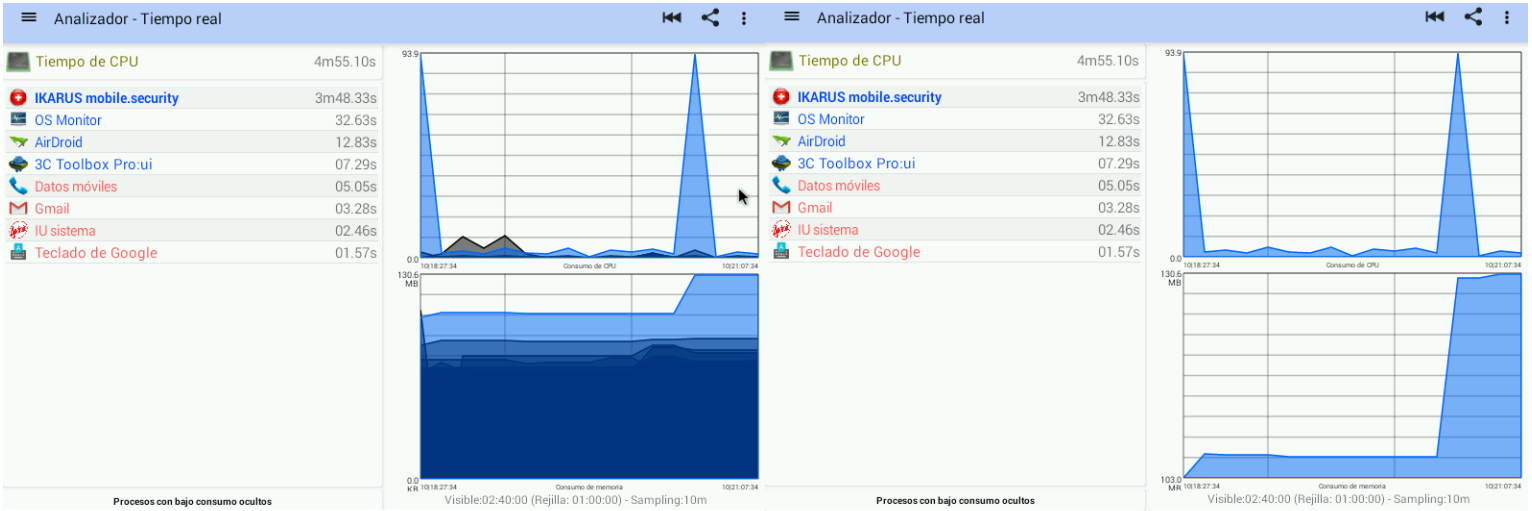


Figura 6.13 – Consumo de CPU y memoria herramienta de IKARUS

Figura 6.14 – Detalle del proceso IKARUS

INTEL SECURITY

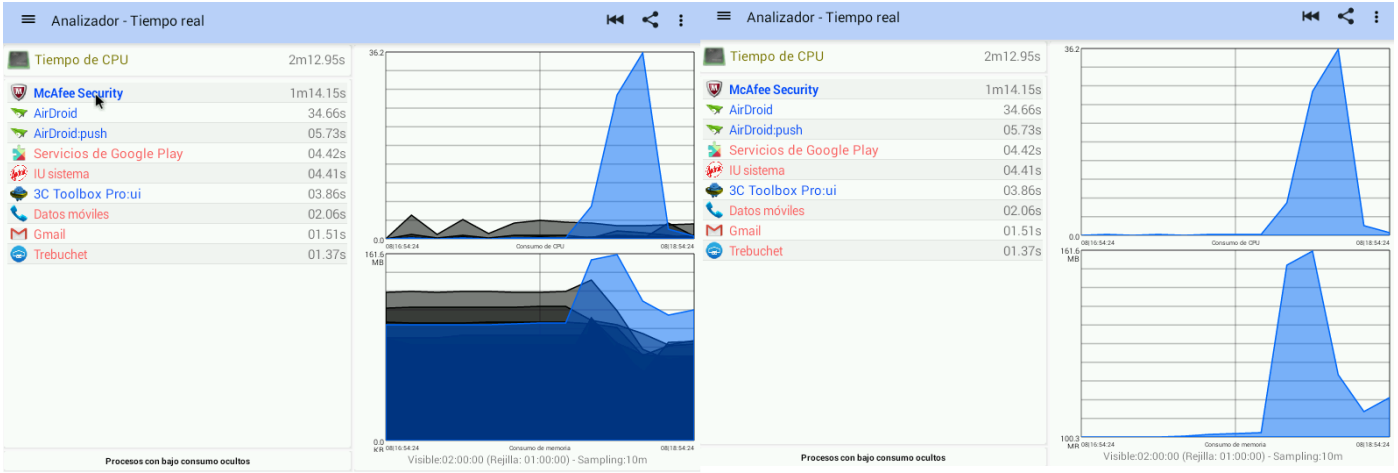


Figura 6.15 – Consumo de CPU y memoria herramienta de Intel Security

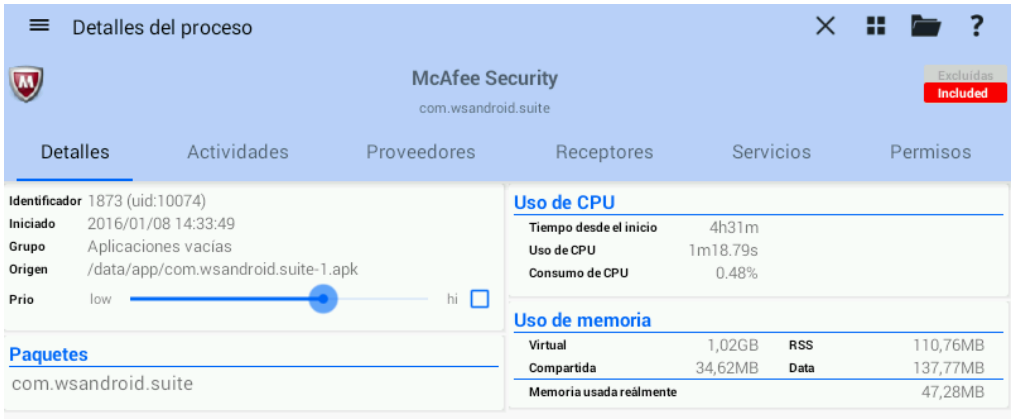


Figura 6.16 – Detalle del proceso Intel Security

Como podemos observar de los resultados, el consumo máximo de CPU del proceso ha sido de un 36,2% y la necesidad de memoria ha sido de 161,6 MB. Además, en la información que nos ofrece el detalle del proceso vemos que sólo ha usado un 0,48% de CPU desde que está iniciado el proceso y la memoria usada realmente es de 47,28MB.

KASPERSKY

En los resultados obtenidos observamos que el consumo máximo de CPU del proceso ha sido de un 25,2% y la necesidad de memoria ha sido de 95,5 MB. Además, en la información que nos ofrece el detalle del proceso vemos que sólo ha usado un 0,54% de CPU desde que está iniciado el proceso y la memoria usada realmente es de 25,68MB.

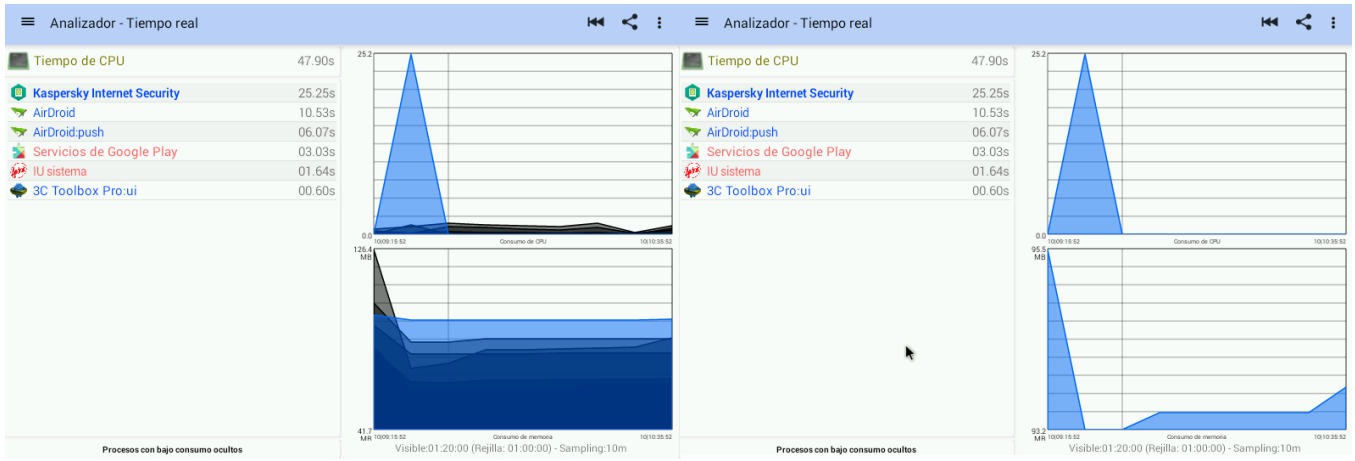


Figura 6.17 – Consumo de CPU y memoria herramienta de Kaspersky

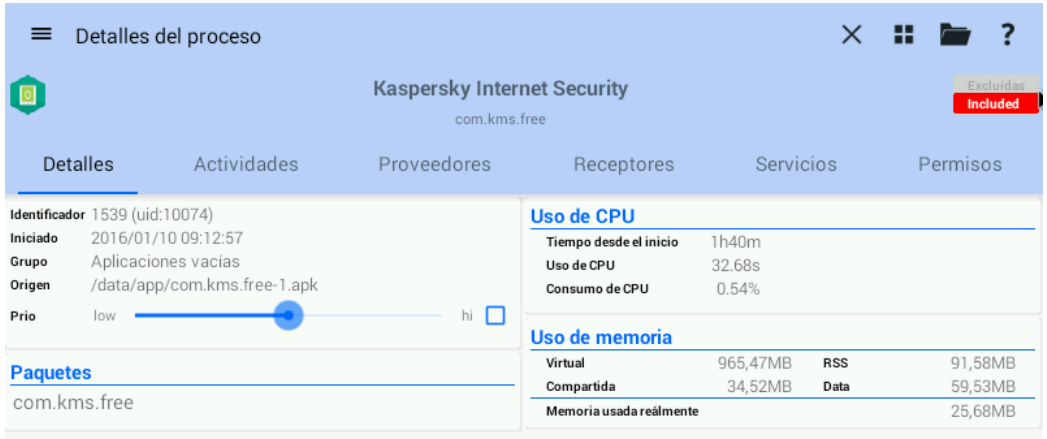


Figura 6.18 – Detalle del proceso Kaspersky

SOPHOS

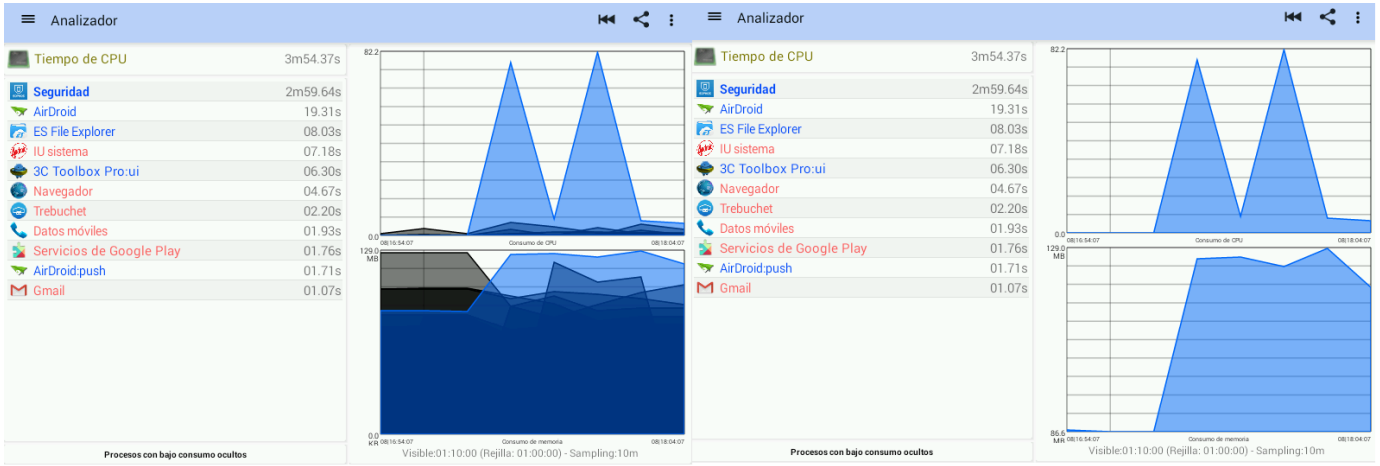


Figura 6.19 – Consumo de CPU y memoria herramienta de SOPHOS

Figura 6.20 – Detalle del proceso SOPHOS

A razón de los resultados obtenidos, el consumo máximo de CPU del proceso ha sido de un 82,2%, pero que hay que tener en cuenta que ha sido puntualmente. Por otra parte, la necesidad de memoria ha sido de 129 MB, valor acorde con el resto de las herramientas analizadas. Además, en la información que nos ofrece el detalle del proceso vemos que sólo ha usado un 1,50% de CPU desde que está iniciado el proceso y la memoria usada realmente es de 32,68 MB.

TREND MICRO

Los resultados nos muestran que el consumo de CPU máximo del proceso ha sido de un 14,4% y la necesidad de memoria ha sido de 113,6 MB. Además, en la información que nos ofrece el detalle del proceso vemos que sólo ha usado un 0,43%% de CPU desde que está iniciado el proceso y la memoria usada realmente es de 29,29MB.

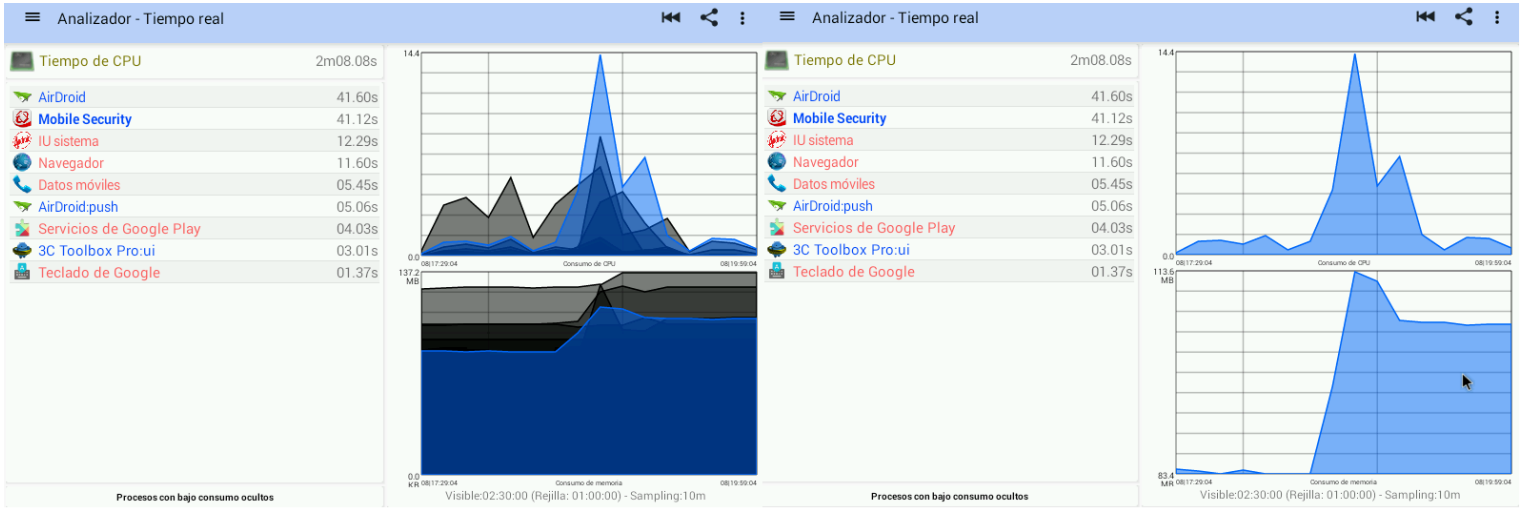


Figura 6.21 – Consumo de CPU y memoria herramienta de Trend Micro

Figura 6.22 – Detalle del proceso Trend Micro

6.2.2. CONSUMO DE BATERIA

Los resultados que nos revela la herramienta de monitorización, han sido obtenidos, tal como se comentó en la metodología de análisis consumo de batería (punto 3.3), con un consumo de la batería de más del 50%. Es decir, se ha enchufado el dispositivo para cargar la batería y cuando ha estado cargada al 100% se desenchufa. En ese momento la herramienta pone los contadores a 0 y se han hecho diferentes operaciones de funcionamiento hasta conseguir que la misma quede por debajo del 50%. Entonces es cuando hemos realizado las capturas de pantalla del dispositivo que se muestran para cada fabricante.

AVAST

Figura 6.23 – Consumo de batería ejecutando la herramienta de AVAST

Observamos, en la captura de pantalla de la izquierda que el 57% del consumo de la batería ha sido debido a las aplicaciones, y haciendo un despliegue del campo “Uso de aplicaciones” la herramienta de monitorización nos muestra, en la captura de pantalla de la derecha que la aplicación Avast Mobile Security ha consumido un 11,5% de la batería.

AVG

Figura 6.24 – Consumo de batería ejecutando la herramienta de AVG

De las capturas de pantalla se desprende que el 52% del consumo de la batería ha sido debido a las aplicaciones, y desplegando el campo “Uso de aplicaciones”, la herramienta de monitorización nos muestra que la aplicación Antivirus ha consumido el 2,2% de la batería.

AVIRA

Figura 6.25 – Consumo de batería ejecutando la herramienta de AVIRA

En las capturas de pantalla vemos como el 53% del consumo de la batería ha sido debido a las aplicaciones y, haciendo un despliegue del campo “Uso de aplicaciones” la herramienta de monitorización nos presenta que la aplicación Avira ha consumido el 2,9% de la batería.

BITDEFENDER

Figura 6.26 – Consumo de batería ejecutando la herramienta de BitDefender

En la captura de pantalla de la izquierda vemos que el 60% del consumo de la batería ha sido debido a las aplicaciones, y haciendo un despliegue del campo “Uso de aplicaciones” la herramienta de monitorización nos revela, en la captura de pantalla de la derecha que la aplicación Antivirus ha consumido el 0,9% de la batería.

ESET

Figura 6.27 – Consumo de batería ejecutando la herramienta de ESET

Observamos que el 51% del consumo de la batería ha sido debido a las aplicaciones y, desplegando el campo “Uso de aplicaciones” en la herramienta de monitorización, vemos que la aplicación Mobile Security & Antivirus ha consumido el 0,8% de la batería.

IKARUS

Figura 6.28 – Consumo de batería ejecutando la herramienta de IKARUS

La captura de pantalla de la izquierda nos muestra que el 57% del consumo de la batería ha sido debido a las aplicaciones y, desplegando el campo “Uso de aplicaciones” en la herramienta de monitorización, se pone de manifiesto que la aplicación IKARUS mobile.security ha consumido el 5,2% de la batería.

INTEL SECURITY

Figura 6.29 – Consumo de batería ejecutando la herramienta de Intel Security

Observamos que el 69% del consumo de la batería ha sido debido a las aplicaciones, y haciendo un despliegue del campo “Uso de aplicaciones” la herramienta de monitorización nos muestra, en la captura de pantalla de la derecha, que la aplicación McAfee Security ha consumido un 14,6% de la batería.

KASPERSKY

Figura 6.30 – Consumo de batería ejecutando la herramienta de KASPERSKY

Comprobamos, en la captura de pantalla de la izquierda, como el 60% del consumo de la batería ha sido debido a las aplicaciones y, si desplegamos el campo “Uso de aplicaciones” en la herramienta de monitorización, la herramienta nos ofrece información sobre el hecho que la aplicación Kaspersky Internet Security ha consumido el 3,1% de la batería.

SOPHOS

Figura 6.31 – Consumo de batería ejecutando la herramienta de SOPHOS

El 61% del consumo de la batería ha sido debido a las aplicaciones, tal como nos muestra la figura de la derecha y, despliegando el campo “Uso de aplicaciones” en la herramienta de monitorización vemos que la aplicación Seguridad ha consumido el 14,5% de la batería.

TREND MICRO

Figura 6.32 – Consumo de batería ejecutando la herramienta de Trend Micro

De las capturas de pantalla se desprende que el 51% del consumo de la batería ha sido debido a las aplicaciones, y haciendo un despliegue del campo “Uso de aplicaciones” la herramienta de monitorización nos muestra que la aplicación Mobile Security ha consumido el 3,7% de la batería.

6.2.3. RESUMEN EVALUACIÓN RENDIMIENTO Y CONSUMO DE BATERÍA

Finalmente, presentamos la Tabla 6.1 dónde se muestran los resultados obtenidos en las pruebas realizadas en cuanto a la evaluación del impacto de rendimiento y consumo de batería de las diferentes herramientas de prevención evaluadas.

Hay que tener en cuenta, que como hemos visto en las gráficas que nos ofrecían los resultados de impacto de rendimiento (en el apartado 6.2.1), los valores apuntados en la tabla son los picos de utilización de CPU y de consumo de memoria, obtenidos cuando

hemos realizado análisis manuales con las diferentes herramientas. El consumo de estos recursos de hardware en condiciones normales ha sido muy inferior.

PRODUCTO	CONSUMO MÁXIMO DE CPU	CONSUMO MÁXIMO DE MEMORIA	CONSUMO DE BATERIA
AVAST - Mobile Security	8,1%	134 MB	11,5%
AVG - Antivirus	22,5%	138 MB	2,2%
Avira - Antivirus Security	42,7%	107,9 MB	2,9%
Bitdefender - Mobile Security	58,7%	125,8 MB	0,9%
ESET - Mobile Security	31,1%	125 MB	0,8%
IKARUS - mobile.security	94,0%	130 MB	5,2%
Intel Security-McAfee Security	36,2%	161,6 MB	14,6%
Kaspersky - Internet Security	25,2%	95,5 MB	3,1%
Sophos - Mobile Security	82,2%	129 MB	14,5%
Trend Micro - Mobile Security	14,4%	113,6 MB	3,7%

Tabla 6.1 – Resumen evaluación rendimiento y consumo de batería.

A la vista de estos resultados concluimos lo siguiente:

- **Consumo máximo de CPU**: La solución de Ikarus y Sophos han sido las que han dado los resultados más altos (cuando hemos forzado un análisis manual) con unos valores de 94% y 82,2% respectivamente. Las que menos CPU han consumido han sido las herramientas de AVAST y Trend Micro, con un 8,1% y un 14,4 %. Todo el resto de herramientas se han mantenido entre un 20% y un 60% de utilización de CPU.
- **Consumo máximo de memoria**: Vemos como la herramienta que más memoria requiere es la de Intel Security con 161,6 MB seguida de la de AVG con 138 MB y AVAST con 134 MB. Las que menos memoria han necesitado han sido las de Kaspersky, con 95,5 MB, la de Avira, con 107,9 MB y la de Trend Micro, con 113,6 MB. Todo el resto de herramientas se han mantenido en un abanico entre 125 MB y 140 MB.
- **Consumo de batería**: La solución que menos batería ha consumido, hasta unos niveles irrelevantes, ha sido la de ESET, seguida muy de cerca por Bitdefender. En ambos casos lo que han usado ha estado por debajo del 1%. A continuación aparecen AVG, Avira y Kaspersky con un 2,2%, 2,9% y un 3,1% respectivamente, valores muy pequeños que no se tienen que notar en un uso normal del dispositivo. Los que mayores consumos han requerido (que sin ser valores muy elevados, sí pueden ser percibidos en cuanto a la durabilidad de la batería) han sido Intel Security, Sophos y AVAST.

7. GUÍA DE BUENAS PRÁCTICAS

A continuación presentamos una recopilación de buenas prácticas, para estar protegido frente al malware en Android.

- Tener Android a la última versión posible para el dispositivo en cuestión. Hemos visto cómo a lo largo de las diferentes versiones aparecidas de la plataforma Android, han ido apareciendo mejoras en el ámbito de la seguridad que nos protegen contra el malware, por lo tanto, nos interesa estar actualizados.
- No *rootear* el dispositivo. Si lo hacemos perderemos la seguridad que ofrece el *sandboxing* y una app se podría ejecutar con los mismos permisos del sistema.
- Solamente instalar apps de Play Store. Hemos mostrado, en el punto 2.3.4, como Google ha implementado medidas para verificar las aplicaciones que consideramos interesantes para poder instalarnos apps de este repositorio con seguridad y garantías.
- No permitir instalar apps de fuentes desconocidas. Desmarcar la opción (en Ajustes→ Sistema → Seguridad → Fuentes desconocidas) que permite instalar apps desde otro lugar que no sea el repositorio de Google, ya que esto evitará que alguna app maliciosa instale en modo silencioso otras apps.
- Si se ha de instalar alguna app de forma manual.
    - Tener activado, la funcionalidad de verificación de aplicaciones (Ajustes → Sistema → Seguridad → Verificar aplicaciones). Disponible a partir de la versión Jelly Bean (4.2).
    - Previamente pasar la app por alguna plataforma (como VirusTotal) que nos permita comprobar que realmente está libre de malware.
- Instalar una herramienta de prevención contra el malware. En este TFM hemos comprobado que son efectivas, con un porcentaje muy elevado de detecciones, y que no perjudican al rendimiento del sistema. Con esta base consideramos que no está de más tener una herramienta instalada ya que contra más medidas de seguridad apliquemos más protegidos estaremos.
- Revisar los permisos de las aplicaciones cuando las instalemos y cuando las actualicemos. Hemos de comprobar que realmente los permisos que requiere la app se limiten a lo que la aplicación pretende hacer. Por poner un ejemplo bastante trivial, si una aplicación es una linterna, no tiene sentido que pida permisos de navegación ya que el único permiso que necesita es acceder al flash del dispositivo.
- Eliminar aplicaciones innecesarias. Aplicaciones que en su día se instalaron por algún motivo, si no son necesarias mejor no tenerlas instaladas. Cuantas más aplicaciones se

tengan instaladas, mayor será el riesgo de que alguna pueda ser maliciosa.

En cuanto a otras cuestiones que no son tan específicas de la plataforma Android pero que nos proporcionaran un grado adicional de seguridad tenemos:

- Tener precaución y cautela cuando recibamos, ya sea por correo electrónico o sms, mensajes de desconocidos (o no tan desconocidos) que puedan contener enlaces. Como recomendación general, no ir a esos enlaces ya que a través de ellos se nos puede infectar el dispositivo.
- Deshabilitar, a través del operador de telefonía móvil, la posibilidad de enviar mensajes SMS a servicios Premium. Esto nos ahorrará dinero en caso que se nos haya instalado un malware que tiene este tipo de comportamiento. Hemos visto que este tipo de malware ha sido el más extendido en los últimos años.
- Hacer copias de seguridad de la información contenida en el dispositivo. Esta recomendación es debida a que si nos "secuestran" el mismo o nos cifran la información, no tendremos que pagar para recuperarla ya que la tendremos nosotros y haciendo un formateo del dispositivo podremos volver a tenerlo operativo.
- Utilizar navegación con seguridad en el transporte (HTTPS) siempre que sea posible e ir a páginas de confianza.
- En cuanto a la seguridad física es imprescindible poner algún tipo de bloqueo de pantalla y PIN a la tarjeta SIM.

Como reflexión final, hemos de tener presente que en el dispositivo tenemos información personal y confidencial con la que ciberdelincuentes pretenden hacer negocio. Además el propio dispositivo ofrece interesantes opciones para que los atacantes puedan obtener beneficios monetarios a costa nuestra. Por tanto, hemos de estar prevenidos y alertas ya que el mejor aliado para proporcionarnos seguridad somos nosotros mismos y el sentido común.

8. CONCLUSIONES Y TRABAJO FUTURO

Como conclusiones a las que llegamos a raíz del trabajo realizado destacamos que el malware en Android existe, que es un problema grave y que, debido al éxito de la plataforma, no para de crecer, en cuanto a cantidad y variedad. El malware puede ser de muchos tipos (virus, troyanos, gusanos, caballos de troya, etc…) pero todos ellos tienen en común que, como último fin, pretenden la obtención de beneficios económicos por parte de

los ciberdelincuentes.

Google, propietaria de Android, ha implementado multitud de medidas de seguridad para proteger a los usuarios del malware y consideramos que son adecuadas (en muchos aspectos, aunque en otros no tanto) para proporcionar un entorno seguro para un usuario que haga un uso normal del dispositivo. Hay desplegadas, a todos los niveles, medidas de seguridad que pueden prevenir gran parte de las infecciones por parte de los dispositivos que utilizan Android. Estas son:

- La propia arquitectura de cómo funciona el dispositivo en cuanto a la ejecución de las aplicaciones, estamos hablando de sandboxing y permisos de las aplicaciones. Estas son adecuadas, siempre y cuando el usuario no *rootee* el dispositivo y se fije en los permisos que cada aplicación le solicita cuando se instala.
- Verificar la procedencia de las aplicaciones con certificados. Esta medida podemos pensar que no es muy efectiva para las primeras instalaciones, ya que los certificados que admite pueden ser auto-firmados, pero para las actualizaciones sí que será útil ya que se comprueba que el certificado de quien ha instalado la aplicación sea el mismo de quien la quiere actualizar.
- Verificación de las aplicaciones tanto de las que hay publicadas en Play Store, a través de la tecnología Google Bouncer, como de las que se instalen en el dispositivo. Ésta última funcionalidad está disponible a partir de la versión Jelly Bean.
- Política para desarrolladores de Play Store. Más clara y restrictiva para que los desarrolladores tengan presente que no se les permitirá subir aplicaciones que vulneren esta política.
- Eliminación remota de aplicaciones. Permite a Google desinstalar de forma masiva apps que considere que son maliciosas en caso que se detectara un problema masivo.
- Cifrado de datos. Para en caso que se llegase a infectar el dispositivo con algún malware que roba información confidencial proteger la confidencialidad de la misma.

Por otra parte, hay empresas que han desarrollado todo un abanico de herramientas (suites de seguridad) que ofrecen funcionalidades adicionales para proteger los dispositivos con Android frente al malware, ya provenga de apps, ficheros, entorno web, mensajes SMS o correos electrónicos. Además, estas herramientas pueden ofrecer como funcionalidades adicionales protección antirrobo, control parental, copia de seguridad, auditoria de aplicaciones instaladas o bloqueo de SMS y llamadas.

En base a nuestra evaluación de diez herramientas de prevención, hemos concluido que son efectivas pero no infalibles. De todas formas, consideramos que es necesario tenerlas

instaladas, ya que agregan una capa complementaria de seguridad, y aportan un obstáculo adicional para que el dispositivo no sea comprometido. Sobre todo, estimamos que es sumamente recomendable para los usuarios que hagan un uso intensivo del dispositivo y se instalen apps de otras fuentes que no sean Play Store, que accedan a bancos, que trabajen con información empresarial, etc., además, hay que tener en cuenta que no siempre se siguen las mejores prácticas en cuanto a la seguridad se refiere.

En cuanto al impacto sobre el rendimiento del dispositivo por el hecho de tener medidas de prevención adicionales a las planteadas por Google, podemos concluir, en base a las pruebas realizadas que, si nos centramos en el consumo de CPU y utilización de memoria no se producirá ningún perjuicio si éstas se ejecutan sobre dispositivos actuales (ya sean de gama alta o media) ya que disponen de procesadores de varios cores y una memoria RAM de 1 GB mínimo (hay hasta con 4 GB). Es posible que puedan haber problemas en dispositivos más antiguos en los cuales el hardware sea más limitado, ya que si alguno tiene 512MB de RAM por ejemplo, hemos de tener presente, basándonos en los test, que la herramienta que menos memoria consume, cuando se realiza un análisis de forma manual o programado, está sobre los a 100MB.

Centrándonos en la evaluación realizada para medir el consumo de batería que las herramientas de prevención utilizan, concluimos que, dependiendo de la herramienta sí que se puede percibir, en cierta medida, una menor autonomía, ya que hay alguna herramienta que ha consumido casi el 15% de la batería. De todas formas, en la mitad de las que hemos testeado, el consumo de la batería por parte de la aplicación ha estado inferior al 3% y en dos casos incluso hemos obtenido resultados por debajo del 1%, consumo realmente insignificante.

Respecto a los trabajos futuros que se podrían realizar para continuar y profundizar sobre la temática del presente TFM en cuanto al malware en Android y medidas de prevención, podemos pensar en:

- Seguir la evolución de las medidas de seguridad que Google seguirá implementando para la plataforma Android.
- Continuar analizando el software malicioso que va a seguir apareciendo, en cuanto a tipologías, familias, formas de infección, funcionalidades, etc.
- Automatizar trabajos para comprobar la efectividad de las herramientas de prevención. Es decir, realizar apps o scripts para poder probar más herramientas de prevención con más muestras.
- Probar durante periodos de tiempo largos el impacto en el rendimiento y en el consumo

de la batería.

- Hacer los test de rendimiento y consumo de batería en todo tipo de dispositivos, no solo en entorno virtual o de gama alta, que vayan más justos de recursos hardware.
- Añadir en cuanto a la evaluación de las herramientas de prevención un apartado de usabilidad.

9. REFERENCIAS BIBLIOGRÁFICAS Y ENLACES

[1] https://es.wikipedia.org/wiki/Andy_Rubin#Retiro_de_la_Compa.C3.B1.C3.ADa_Google. Accedida por última vez el 24/01/2016.

[2] http://www.openhandsetalliance.com/press_110507.html. Accedida por última vez el 24/01/2016.

[3] http://www.openhandsetalliance.com/press_111207.html. Accedida por última vez el 24/01/2016.

[4] https://www.idc.com/getdoc.jsp?containerId=prUS25804315. Accedida por última vez el 24/01/2016.

[5] https://en.wikipedia.org/wiki/Android_version_history. Accedida por última vez el 24/01/2016.

[6] http://www.cnet.com/es/noticias/android-lollipop-instalado-1-cada-4-dispositivos-marshmallow-adopcion/. Accedida por última vez el 24/01/2016.

[7] http://www.cnet.com/es/noticias/google-1400-millones-usuarios-android/. Accedida por última vez el 24/01/2016.

[8] https://developer.android.com/intl/es/about/dashboards/index.html. Accedida por última vez el 24/01/2016.

[9] https://sites.google.com/site/swcuc3m/home/android/generalidades/2-2-arquitectura-de-android. Accedida por última vez el 24/01/2016.

[10] http://androideity.com/2011/07/04/arquitectura-de-android/. Accedida por última vez el 24/01/2016.

[11] http://www.hermosaprogramacion.com/2014/08/aprendiendo-la-arquitectura-de-android/. Accedida por última vez el 24/01/2016.

[12] https://es.wikipedia.org/wiki/GNU_General_Public_License. Accedida por última vez el 24/01/2016.

[13] http://www.cnet.com/es/analisis/google-android-6-0-marshmallow/. Accedida por última vez el 24/01/2016.

[14] https://es.wikipedia.org/wiki/Aislamiento_de_procesos_(inform%C3%A1tica).

Accedida por última vez el 24/01/2016.

[15] https://www.nsa.gov/research/selinux/. Accedida por última vez el 24/01/2016.

[16] Guía de Seguridad de las TIC (CCN-STIC-453B); Título: "Seguridad de dispositivos Móviles: Android 4.x"; URL: https://www.ccn-cert.cni.es/pdf/guias/series-ccn-stic/guias-de-acceso-publico-ccn-stic/827-ccn-stic-453b-seguridad-en-android-4-x.html. Accedida por última vez el 24/01/2016.

[17] https://developer.android.com/intl/es/reference/android/Manifest.permission.html. Accedida por última vez el 24/01/2016.

[18] https://developer.android.com/intl/es/reference/android/Manifest.permission_group.html. Accedida por última vez el 24/01/2016.

[19] http://blog.trendmicro.com/trendlabs-security-intelligence/a-look-at-google-bouncer/. Accedida por última vez el 24/01/2016.

[20] https://developer.android.com/intl/es/tools/help/adb.html. Accedida por última vez el 24/01/2016.

[21] http://officialandroid.blogspot.com.es/2014/04/expanding-googles-security-services-for.html. Accedida por última vez el 24/01/2016.

[22] http://andro4all.com/2014/03/google-aumenta-seguridad-en-google-play-para-usuarios. Accedida por última vez el 24/01/2016.

[23] http://googlemobile.blogspot.com.es/2011/03/update-on-android-market-security.html. Accedida por última vez el 24/01/2016.

[24] http://www.oracle.com/technetwork/java/embedded/javame/index.html. Accedida por última vez el 24/01/2016.

[25] http://www.hermosaprogramacion.com/2014/08/aprendiendo-la-arquitectura-de-android/

[26] "Descubriendo la anatomía de una aplicación sobre Android.pdf" – URL: http://www.lamjol.info/index.php/NEXO/article/view/685/849.

[27] https://sites.google.com/site/swcuc3m/home/android/generalidades/aplicacionesandroid. Accedida por última vez el 24/01/2016.

[28] https://www.gartner.com/newsroom/id/2408515. Accedida por última vez el 24/01/2016.

[29] https://www.f-secure.com/documents/996508/1030743/Mobile_Threat_Report_Q3_2013.pdf. Accedida por última vez el 24/01/2016.

[30] https://www.f-secure.com/documents/996508/1030743/Mobile_Threat_Report_Q1_2014.pdf. Accedida por última vez el 24/01/2016.

[31] http://www.infochannel.com.mx/android-acapara-965--del-malware-rastreado-en-moviles-fortinet. Accedida por última vez el 24/01/2016.

[32] https://www.juniper.net/us/en/local/pdf/additional-resources/jnpr-2012-mobile-threats-report.pdf. Accedida por última vez el 24/01/2016.

[33] "Symantec Internet security threat report 2015 – appendices" (pág. 25) – URL: https://www.symantec.com/security_response/publications/threatreport.jsp. Accedida por última vez el 24/01/2016.

[34] https://www.ccn-cert.cni.es/seguridad-al-dia/noticias-seguridad/3316-informe-del-estado-de-la-ciberseguridad-2015.html. Accedida por última vez el 24/01/2016.

[35] https://www.educacion.gob.es/teseo/mostrarRef.do?ref=957612#. Accedida por última vez el 24/01/2016.

[36] Gary McGraw y Greg Morrisett. Attacking malicious code: A report to the infosec research council. IEEE Software, 17:33– 41, 2000.

[37] M. Christodorescu. Behavior-based malware detection. Tesis Doctoral, University of Wisconsin-Madison, 2007.

[38] M. Dalla Preda. Code Obfuscation and Malware Detection by Abstract Interpretation. Tesis Doctoral, Universit'a degli Studi di Verona, 2007.

[39] Apuntes asignatura Delitos informáticos del Master de Seguridad Informática de UNIR.

[40] M. E. Karim, A. Walenstein, A. Lakhotia, y L. Parida. Malware phylogeny generation using permutations of code. Journal in Computer Virology, 1(1):13–23, 2005.

[41] http://forensics.spreitzenbarth.de/android-malware/. Accedida por última vez el 24/01/2016.

[42] Dissecting Android Malware Characterization and Evolution.pdf – URL: http://www.ieee-security.org/TC/SP2012/papers/4681a095.pdf. Accedida por última vez el 24/01/2016.

[43] https://static.googleusercontent.com/media/source.android.com/es//security/reports/Google_Android_Security_2014_Report_Final.pdf. Accedida por última vez el 24/01/2016.

[44] https://www4.symantec.com/mktginfo/whitepaper/ISTR/21347932_GA-internet-security-threat-report-volume-20-2015-social_v2.pdf. Accedida por última vez el 24/01/2016.

[45] http://www.androidcentral.com/smobile-systems-provide-security-android. Accedida por última vez el 24/01/2016.

[46] https://es.wikipedia.org/wiki/Virus_de_telefon%C3%ADa_m%C3%B3vil#Historia. Accedida por última vez el 24/01/2016.

[47] http://www.smh.com.au/digital-life/consumer-security/mobile-antivirus-not-needed-

google-20140702-zsthl.html. Accedida por última vez el 24/01/2016.

[48] http://www.adslzone.net/2014/10/18/es-necesario-tener-un-antivirus-en-el-smartphone-o-tablet/. Accedida por última vez el 24/01/2016.

[49] http://www.xataka.com/moviles/antivirus-android-eficaces-o-vulnerables. Accedida por última vez el 24/01/2016.

[50] http://www.muycomputer.com/2014/07/16/antivirus-android-funcionan. Accedida por última vez el 24/01/2016.

[51] http://rootear.com/android/son-necesarios-antivirus-android. Accedida por última vez el 24/01/2016.

[52] http://tecnologia.elpais.com/tecnologia/2012/09/10/actualidad/1347276010_539192.html. Accedida por última vez el 24/01/2016.

[53] https://www.virustotal.com/ca/documentation/mobile-applications/. Accedida por última vez el 24/01/2016.

[54] http://www.androidjefe.com/instalar-antivirus-android-no-necesario-google/. Accedida por última vez el 24/01/2016.

[55] http://www.xatakamovil.com/seguridad/sirven-para-algo-los-antivirus-para-telefonos-moviles. Accedida por última vez el 24/01/2016.

[56] http://hdl.handle.net/11059/2513. Accedida por última vez el 24/01/2016.

[57] Tesis – "Nuevo Enfoque para la detección de malware basado en métodos de recuperación de información" – URL: https://www.educacion.gob.es/teseo/mostrarRef.do?ref=957612#. Accedida por última vez el 24/01/2016.

[58] http://www.android-x86.org/download. Accedida por última vez el 24/01/2016.

[59] http://www.android-x86.org/. Accedida por última vez el 24/01/2016.

[60] http://www.3c71.com/android/?q=node/916. Accedida por última vez el 24/01/2016.

[61] https://www.airdroid.com/es-es/. Accedida por última vez el 24/01/2016.

[62] https://play.google.com/store/apps/details?id=com.gsamlabs.bbm.pro. Accedida por última vez el 24/01/2016.

[63] https://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/reports/rpt-repeating-history.pdf. Accedida por última vez el 24/01/2016.

[64] https://www.f-secure.com/documents/996508/1030743/Mobile+Threat+Report+Q4+2012.pdf. Accedida por última vez el 24/01/2016.

[65] https://www.f-secure.com/documents/996508/1030743/Threat_Report_H1_2013.pdf. Accedida por última vez el 24/01/2016.

[66] https://www.f-secure.com/documents/996508/1030743/Threat_Report_H2_2013.pdf.

Accedida por última vez el 24/01/2016.

[67] https://www.f-secure.com/documents/996508/1030743/Mobile_Threat_Report_Q3_2013.pdf. Accedida por última vez el 24/01/2016.

[68] https://www.pulsesecure.net/lp/mobile-threat-report-2014/?receipt_id=true. Accedida por última vez el 24/01/2016.

[69] www.virustotal.com. Accedida por última vez el 24/01/2016.

[70] https://www.av-test.org y http://www.av-comparatives.org/. Accedidas por última vez el 24/01/2016.

[71] https://www.av-test.org/es/noticias/news-single-view/las-mejores-aplicaciones-para-android-25-guardianes-sometidos-a-una-prueba-de-larga-duracion/. Accedida por última vez el 24/01/2016.

[72] http://www.av-comparatives.org/wp-content/uploads/2015/09/avc_mob_2015_en.pdf. Accedida por última vez el 24/01/2016.

9.1. ORÍGENES DE FIGURAS

[a] https://developer.android.com/intl/es/about/dashboards/index.html

[b] https://source.android.com/devices/tech/security/

[c] http://hiqes.com/android-security-part-1/

[d] http://show.docjava.com/posterous/file/2012/12/10222640-The_Dalvik_Virtual_Machine.pdf

[e] https://www.gartner.com/newsroom/id/2408515

[f] “Symantec Internet security threat report 2015 – appendices” (pág. 25)

[g] https://static.googleusercontent.com/media/source.android.com/es//security/reports/Google_Android_Security_2014_Report_Final.pdf

[h] https://www4.symantec.com/mktginfo/whitepaper/ISTR/21347932_GA-internet-security-threat-report-volume-20-2015-social_v2.pdf

[i] https://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/reports/rpt-repeating-history.pdf

10. ANEXOS

ANEXO I - FAMILIAS DE MALWARE PARA ANDROID

Nombre	Descripción	Capacidades
AccuTrack	Esta aplicación convierte un teléfono inteligente Android en un rastreador GPS	3 5
Ackposts	Este troyano roba información de contactos desde el dispositivo comprometido y los carga en un servidor remoto.	6
Acnetdoor	Este troyano abre una puerta trasera en el equipo infectado y envía la dirección IP a un servidor remoto.	2 6
Adsms	Se trata de un troyano al que se le permite enviar mensajes SMS. El canal de distribución de este malware es a través de un mensaje SMS con el enlace de descarga.	3 4
Airpush/StopSMS	Airpush es una, muy agresiva, red de anuncions (Ad-Network).	3 8
AnServer/Answerbot	Abre una puerta trasera en los dispositivos Android y es capaz de robar información personal que será subida a un servidor remoto.	6
Antares/Antammi	Este es un troyano que roba información personal del dispositivo infectado.	6
Arspam	Este malware representa la primera etapa de la piratería por motivos políticos (hacktivismo) en plataformas móviles.	4
AVPass	Esta familia de malware intenta detectar y eludir las herramientas de seguridad de Android (como aplicaciones antivirus) instaladas en el dispositivo infectado. Posteriormente, la aplicación trata de obtener datos confidenciales y recibe comandos adicionales a través de SMS.	1 2 5 6
BackFlash/Crosate	Esta aplicación maliciosa instala un plugin de Flash falso que se registra a sí mismo como administrador del dispositivo y roba de información sensible.	1 6
Badaccents	Este malware pretende descargar una copia de "La Entrevista", pero en lugar de ello, instala un troyano bancario de dos etapas en los dispositivos víctimas.	9
Badnews	Una vez activado, BadNews consulta al servidor de C&C si hay nuevas instrucciones a la vez que va enviando información confidencial, como el número de teléfono del dispositivo o IMEI al servidor.	1 3 6
BankBot	Este malware intenta robar información confidencial de los usuarios, concretamente información de autenticación de las cuentas bancarias de los dispositivos infectados.	1 6 8
Basebridge	Envía datos confidenciales (SMS, IMSI, IMEI) a un servidor remoto.	1 6
BeanBot	Se trata de un troyano que puede enviar mensajes de SMS y que es controlado por un servidor de C&C.	1 4 6
Beita	Roba información de contactos almacenados en el dispositivo infectado.	6
Binv	Este malware es un clásico Toryano de Banda que se dirige a los usuarios brasileños de los dispositivos Android.	1 3 6 9
BgServ	Obtiene información del teléfono del usuario (IMEI, número de teléfono, etc.). La información se sube a una URL específica.	1 2 3 6
Biige	Este spyware registra las llamadas, mensajes SMS, ubicación, etc... Y los envía a un servidor remoto.	5 6
Booster	Esta aplicación roba información personal y envía estos datos a un servidor remoto.	6
Boxer	Este troyano envía mensajes SMS a números de tarificación adicional nominal.	4
Cajino	Este malware es una RATA clásica que intenta robar información sensible. Lo que hace esta muestra especial es que utiliza el servicio push de Baidu para la comunicación.	1 6
Carberp	Trata de robar códigos confidenciales bancarios de autenticación (mensajes mTAN) enviados al dispositivo infectado.	6 9
Cawitt	Esta aplicación roba información personal y carga estos datos a un servidor remoto.	6
Cellspy	Esta aplicación es un rastreador.	5
Chulli	Esta familia de malware ha sido utilizado dentro de ataque dirigido. La cuenta de correo electrónico de un activista tibetano de alto perfil fue hackeado y utilizado para enviar ataques dirigidos a otros activistas y defensores de los derechos humanos. Después que un dispositivo móvil se infecta, se conecta a un servidor de C&C y espera órdenes de SMS para enviar de datos sensibles a este servidor.	1 5 6
Code4hk/xRAT	la geolocalización de la víctima, así como grabaciones de voz. La muestra maliciosa se propaga a través de mensajes de WhatsApp.	1 6
Coogos	Troyano de puerta trasera que tiene la capacidad para recibir una conexión remota desde un hacker malicioso y llevar a cabo acciones contra el sistema comprometido.	1 2
CopyCat	Es una red de publicidad agresiva y maliciosa. El objetivo principal es generar ingresos.	3

Nombre	Descripción	Capacidades
Cosha	Esta aplicación monitoriza el dispositivo infectado y envia datos personales a un servidor remoto.	6
Counterclank	Realmente no podría ser considerado como malware, sino un anuncio de red muy agresivo con la capacidad de robar información privada.	3, 4, 6
Crusewind	Intercepta mensajes SMS entrantes y los envía a un servidor remoto que incluyendo información como IMSI e IMEI.	6
Dogowar	Este troyano envía mensajes SMS de spam a todos los contactos.	4
Dougalek	Esta aplicación roba información personal y carga estos datos a un servidor remoto.	6
Dowgin	Es una librería de publicidad que está integrada con diferentes aplicaciones. Es capaz de presentar publicidad en la barra de notificaciones, enviar información de las aplicaciones instaladas, IMEI, versión de kernel, modelo de teléfono, etc. a un servidor remoto, descargar y instalar nuevas aplicaciones.	3, 6, 7, 8
DroidDeluxe	Explota el dispositivo para obtener privilegios de root. Después modifica el permiso de acceso de algunos archivos de bases de datos del sistema y trata de recoger información de la cuenta.	2
DroidDream	Utiliza dos herramientas diferentes (rageagainstthecage y exploid) para rootear el smartphone.	1, 2, 3
DroidDreamLight	Recopila información de un teléfono móvil infectado (dispositivo, IMEI, IMSI, país, la lista de aplicaciones instaladas) y se conecta a varias direcciones URL para subir estos datos.	3, 6
DroidJack/SandoRAT	Este programa malicioso es de características similares a otras ratas Android. Algunas de estas características son las siguientes: Instala APK, ve todos los mensajes en el dispositivo, escucha las conversaciones realizadas en el dispositivo, etc.	1, 6, 7
DroidKungfu	Recoge una variedad de información en el teléfono infectado (IMEI, dispositivo, versión del sistema operativo, etc.). La información recogida se vuelca en un archivo local que se envía a un servidor remoto después.	1, 2, 6
DroidSheep	Esta aplicación puede capturar y secuestrar sesiones web sin encriptar.	8
DSEncrypt	Roba información confidencial (mensajes SMS, certificados y claves privadas, etc.) de los teléfonos inteligentes infectados y carga los datos a un servidor remoto.	6
Extension/Monad	Este troyano es capaz de interceptar las llamadas telefónicas entrantes y salientes, abrir un navegador y visitar sitios web específicos, ejecutar los clics en los anuncios, y es capaz de actualizar su propio código malicioso. Por otra parte, la aplicación correspondiente puede hacer llamadas telefónicas, enviar mensajes SMS y recoge información privada como el historial de llamadas, contactos, localización GPS y el ID de dispositivo, los cuales serán subidos a un servidor remoto.	1, 4, 5, 6
FaceNiff	Esta aplicación puede capturar y secuestrar sesiones web sin encriptar.	8
FakeAngry	Troyano de puerta trasera que tiene la capacidad para recibir una conexión remota desde un hacker malicioso y llevar a cabo acciones contra el sistema comprometido.	1, 2, 6
FakeApp.AL	Clásico Adware para Android.	3, 8
FakeAV	El malware engaña a los usuarios a pagar por la limpieza de otras infecciones inexistentes en su dispositivo. Además de mostrar los mensajes falsos de la infección, el APK también tiene la funcionalidad para interceptar llamadas telefónicas entrantes y salientes, así como mensajes.	8
FakeBank	Esta aplicación es un caballo de Troya para los dispositivos Android que abre una puerta trasera y roba información del dispositivo comprometido. Además, es capaz de infectar a un PC con Windows conectado y engaña al usuario para intercambiar aplicaciones bancarias legítimas con otras maliciosas.	3, 6, 9, 10
FakeDaum/vmwol	El troyano recoge la siguiente información desde el dispositivo comprometido: mensajes SMS, número de teléfono y el IMEI del dispositivo infectado.	6
FakeDefender	Esta aplicación es un caballo de Troya para los dispositivos Android que muestra falsas alertas de seguridad en un intento de convencer al usuario a comprar una aplicación con el fin de eliminar el malware o de seguridad riesgos inexistentes desde el dispositivo.	7
FakeDoc	Este troyano instala aplicaciones adicionales.	7
FakeFlash	Este troyano redirige al ausuario a través de proxies de pago.	8
FakeInst	El fraudware más común. Estas aplicaciones envían mensajes SMS premium.	4
FakeJobOffer	El malware muestra un mensaje de estafa que intenta hacer creer a las víctimas que han sido seleccionados como candidatos. Con el fin de asegurar su colocación en la empresa, tienen que hacer un depósito en una cuenta bancaria.	6, 9
FakeMarket	El objetivo general de esta aplicación maliciosa es simplemente para aumentar fraudulentamente el número de visitas a cerca de 20 sitios web diferentes en búsqueda de google.	3, 8

Nombre	Descripción	Capacidades
FakeMart	El troyano puede realizar las siguientes acciones mientras se esconde a sí mismo como una aplicación legítima de acceso a mercado negro: Borrar el contenido XMBPSP.xml en preferencia compartida y reconfigurarlo para enviar mensajes SMS premium al 81211 o 81308, configurar el dispositivo para el modo silencioso, borrar SMS recibidos del 81211, etc.	4 6
FakeNefix	Esta aplicación roba credenciales de usuario.	6
FakeNotify	Esta aplicación envía mensajes SMS a servicios premium mientras usa técnicas de ofuscación y antidetección para no ser detectado por las herramientas antivirus.	4
FakePlay	La aplicación se ejecutará en segundo plano, mientras va recopilando actividad SMS y periódicamente enviarla a una dirección de proxy de correo electrónico proxy. Una vez ejecutado, el troyano pide privilegios de Administrador del dispositivo.	1 6
FakePlayer	Envía mensajes SMS a números prefijados.	4
FakeRegSMS	Envía mensajes SMS a números de tarificación adicional y trata de ocultar esta acción de los investigadores de malware mediante el uso de algún tipo de esteganografía.	4
FakeTaoBao	Este malware intenta robar las credenciales de usuario de Taobao y Zhifubao. En combinación con otra aplicación del mismo desarrollador también es capaz de enviar mensajes SMS.	4 6 7
FakeTimer	Envía la información personal a un servidor remoto y abre páginas web pornográficas.	1 6
FakeUpdate/Apkqug	Esta familia de malware actúa como descargador automático para más aplicaciones.	7
FakeVertu	Troyano de SMS dirigido a los consumidores Vertu en Japón. Este troyano recibe todos los mensajes SMS entrantes y los sube a un servidor remoto.	6
Find and Call/Fidall	Envía información personal (libreta de direcciones) a un servidor remoto.	6
Finspy	Este troyano es un componente de un producto comercial de vigilancia que monitorea la actividad del usuario.	1 6
Fjcon	Este troyano se conecta a un servidor de C&C, tiene la capacidad de instalar paquetes adicionales y enviar mensajes SMS premium.	1 4 6 7
Flexispy	Este malware monitoriza las llamadas telefónicas, mensajes SMS, actividad de Internet y localización GPS.	5
Foncy	Este troyano encia mensajes SMS premium.	4
Fonefee/Feejar	Este troyano encia mensajes SMS premium.	4
Fokange/Fokonge	Es un malware que roba información, la cual es subida en un servidor remoto.	6
Gamex	Abre una puerta trasera e instala aplicaciones adicionales.	1 2 7
Gazon	Este malware intenta robar información sensible y muestra anuncios. Se propaga a través de mensajes de WhatsApp y SMS.	1 6
Geinimi	Abre una puerta trasera y transmite información desde el dispositivo (IMEI, IMSI, etc.) a un URL específico.	1 6
GGTracker	Envía varios mensajes SMS a un número premium. También roba información del dispositivo.	4
GingerBreak	Es un root exploit para Android 2.2 y 2.3	2
GingerMaster/GingerBreaker	Escala acceso a root y cosecha datos en los teléfonos inteligentes infectados. Estos datos se envían a un servidor remoto después.	2 6
Godwon	Esta aplicación intenta robar datos de contacto y datos personales de la libreta de direcciones local y de la aplicación Skype.	6
GoldenEagle/GlodEagl	Roba información personal y recibe comandos via SMS	1 6
GoneIn60Seconds	Roba información (mensajes SMS, IMEI, IMSI, etc.) y carga los datos a una URL específica.	6
GPspy	Seguimiento de la ubicación del dispositivo infectado.	5
HeHe	Este troyano roba mensajes de texto e intercepta llamadas telefónicas.	6
HideIcon	Roba información (mensajes SMS, IMEI, IMSI, etc.) desde el smartphone infectado y carga los datos a un servidor remoto. Adicionalmente, muestra anuncios a pantalla completa.	6
HippoSMS	Envía varios mensajes SMS a un número premium y borra los mensajes SMS entrantes de estos números.	4
HongTouTou/Adrd	Roba información que posteriormente carga, a través de un proxy local, a un servidor remoto. Los datos se cifran de antes de enviarlos.	6
Iconosys	Esta aplicación roba datos personales.	6
Imlog	Esta aplicación roba datos personales.	6

Nombre	Descripción	Capacidades
Jifake	Esta aplicación envía mensajes SMS a servicios premium.	4
JollyServ	El troyano puede enviar mensajes SMS a servicios premium, enviar mensajes SMS a todos los contactos del usuario infectado e interceptar mensajes SMS recibidos.	4
Jsmshider/Xsider	Abre una puerta trasera y envía la información a una URL específica.	6
Kidlogger	Este troyano roba información personal y la envía a un servidor remoto.	6
KMIN	Intenta enviar los datos del dispositivo Android a un servidor remoto.	6
Ksapp	Este troyano tiene capacidad para el manejo de conexiones de acceso remoto, realizar ataques de DoS o DDoS, captura de teclado, eliminar archivos u objetos, o terminar procesos.	1 2 8
LeNa	Comunicación con un servidor de C&C, descargar e instalar otras aplicaciones, iniciar la actividad del navegador web, actualización de los binarios instalados, etc...	1 2 6
Lien	como el número de teléfono, SMS entrantes y salientes, y audio grabado. Entonces se hace uso de servidores SMTP para enviar los datos robados al atacante.	1 6
Locker/SLocker Ransomware	Este troyano es el primer CriptoLocker para Android.	11
Loicdos	Este troyano tiene la capacidad de realizar ataques de DoS o DDoS.	8
Loozfon	Este troyano roba datos personales.	6
Lovetrap/Luvrtrap	Envía mensajes SMS a números de tarificación adicional y roba información del smartphone.	4
Luckycat	Abre una puerta trasera y escucha comandos desde un servidor remoto.	1 2
Maistealer	Este troyano roba datos personales.	6
Malap	Otro ladrón de información	6
Mania	Este troyano envía SMS a servicios premium.	4
MMarketPay	Este troyano puede comprar automáticamente aplicaciones en tiedas de aplicaciones de Android chinas.	7
MobiDash	Adware clásico que muestra anuncios de pantalla completa para el usuario.	3
MobileSpy/Godwon	Este troyano roba datos personales.	6
MobileTx	Este troyano roba datos personales y los envía a través de mensajes SMS o HTTP.	6
Mobinauten	Esta aplicación hace el seguimiento de la ubicación del teléfono inteligente infectado.	5
Moghava	Compromete todas las imágenes del teléfono inteligente mediante la fusión con una imagen del ayatolá Jomeini.	8
Nandrobox	Este troyano roba datos personales y elimina ciertos mensajes SMS.	6
Netisend	Recopila información de smartphones infectados y carga los datos a una URL específica.	6
Nickispy	Recopila información de smartphones infectados (IMSI, IMEI, localización GPS, etc.) y carga los datos a una URL específica.	1 5 6
Obad	Una de las más sofisticadas familias de malware hasta 2013. Puede enviar SMS a servicios premium, bajarse otros programas de malware, instalarlos y/o enviarlos a través de Bluetooth. Además permite ejecutar remotamente comandos en la consola.	1 2 4 5 6 7
Oldboot/MouaBad	Gana permisos de root por las vulnerabilidades del sistema y vuelve a grabar la partición del sistema. También intenta ejecutar código malicioso en la primera etapa de arranque del sistema para evitar ser limpiado por aplicaciones Antivirus. Después, algunas versiones de esta familia envían mensajes SMS premium y actuan como bots.	1 2 4
OpFake	El segundo fraudware más común. Estas aplicaciones envían mensajes SMS premium.	4
PDAspy	Este troyano roba datos personales y la información de ubicación.	5 6
Penetho	Esta aplicación es una herramienta de hackeo para descifrar contraseñas WiFi.	8
Photsy/Phopsy	Este malware intenta robar todos los ficheros .jpg y .mp4 archivos de un dispositivo infectado.	6
Pincer	Este malware es capaz de enviar mensajes SMS y realizar otras acciones sobre la base de órdenes que recibe de su servidor remoto.	1 6
Pjapps	Abre una puerta trasera y roba información del dispositivo. Este programa malicioso tiene implementadas capacidades de bot.	1
Placms	Este troyano tiene la capacidad para el manejo de conexiones de acceso remoto, realizar ataques de DoS o DDoS, captura de entradas del teclado, eliminar archivos u objetos, o terminar procesos.	1 2

Nombre	Descripción	Capacidades
Plankton	Este malware tiene la capacidad de comunicarse con un servidor remoto, bajar e instalar otras aplicaciones, enviar mensajes SMS a servicios premium, etc...	1 4 6 7
Podec	Este troyano envía mensajes SMS a números de tarificación adicional y es capaz de eliminar el consejo de sistema que Android muestra el usuario normalmente cuando se realizan envíos a servicios Premiun.	4
PoisonCake	Este malware puede instalarse a sí mismo, descifrar y soltar otras cargas útiles, crear servicios en segundo plano y, es capaz de realizar las siguientes acciones maliciosas: Inyectar com.android.phone, enviar e interceptar SMS, visitar sitios WAP, recopilar la información del teléfono y subirlos a un servidor remoto, etc...	1 4 6 7
Proxy Trojan/NioServ	Este troyano roba datos personales.	6
Qicsomos	Envía mensajes SMS a números premium.	4
Raden	Envía un mensaje SMS a un número premium chino.	3 4
Repane	Ladrón de información.	6
Roidsec/Sinpon	Ladrón de información.	5 6
RootSmart/Bmaster	Este malware se aprovecha de la vulnerabilidad GingerBreak para obtener privilegios de root. Este exploit no está incrustado en la aplicación sino que se descarga de forma dinámica desde un servidor remoto junto con otras aplicaciones maliciosas.	1 2 4 5 6 7
RuFraud	Envía mensajes SMS premium. Esta es la primera aplicación maliciosa de este tipo que se construye especialmente para los países europeos.	3 4
Saiva	Este troyano tiene las capacidades para el manejo de conexiones de acceso remoto, realizar DoS o DDoS, captura de entradas de teclado, eliminar archivos u objetos, o terminar procesos.	1 2
Samsapo	Este malware se propaga a través de mensajes SMS maliciosos y se comunica con un servidor de C&C. Las muestras correspondientes tienen la capacidad de instalar paquetes adicionales y enviar mensajes SMS premium.	1 4 7
SaveMe/SocialPath	Este malware roba mensajes SMS, contactos registros de llamadas, así como la información del dispositivo y la carga a un servidor remoto.	3 6
Scavir	Envía mensajes SMS a números premium.	4
Scipiex	Ladrón de información.	6
SeaWeth	Este troyano tiene las capacidades para el manejo de conexiones de acceso remoto, realizar DoS o DDoS, captura de entradas de teclado, eliminar archivos u objetos, o terminar procesos.	1 2
Selfmite	Este gusano SMS utiliza una plataforma de publicidad legal y de pago por instalación para la monetización. Se está extendiendo a través de mensajes SMS.	1
Skullkey	El troyano se oculta usando la vulnerabilidad de Android Master Key para mantener válida la firma legítima aplicación. Se permite que los atacantes realicen las siguientes acciones: abrir puerta trasera, robar datos sensibles (como el IMEI y número de teléfono) y la envía a un servidor remoto, enviar mensajes SMS premium, etc.	1 4 6
Smack	Subir información de contactos de usuarios, mensajes cortos, registros telefónicos, localización GPS y fecha, oculta su icono e intercepta mensajes cortos específicos.	1 6
SMSpacem	Recopila información del smartphone y carga estos datos a una URL específica. Este malware también envía mensajes SMS.	1 4 6
SMSreg	Registra el smartphone infectado a servicios de pago.	8
SMSilence/SMSCatcher	SMS troyano dirigido a los consumidores de Starbucks en Corea del Sur. Este troyano recibe todos los mensajes SMS entrantes y los sube a un servidor remoto.	6
SMSspy	Troyano bancario con destinatarios los consumidores españoles.	9
SMSsniffer	Envía copias de los mensajes SMS.	4
Sndapps/Snadapps	El malware es capaz de acceder a diversa información del dispositivo: el operador y el país, ID del dispositivo, dirección de correo electrónico y número de teléfono. Sube esta información a un servidor remoto.	6
SpamBot	Envía mensajes de spam SMS. La aplicación obtiene el contenido del mensaje de correo no deseado y el número del receptor a través de un servidor de C&C.	1 4
Spitmo	Es una de las primeras versiones de los troyanos SpyEye para Android, que roba información del smartphone infectado. El troyano también monitorea e intercepta mensajes SMS de los bancos (mensajes mTAN) y los carga a un servidor remoto.	6 9
SPPush	Este malware envía mensajes SMS premium y publica información privada en un servidor remoto.	4 6 7
SpyBubble	Este troyano roba datos personales.	6
SpyOO	Este troyano registra y roba datos personales.	6

Nombre	Descripción	Capacidades
Ssucl	Este troyano es el primer troyano para Android que es capaz de infectar a un PC con Windows conectado. Además, es capaz de enviar mensajes SMS, habilitar el Wi-Fi, recopilar información sobre el dispositivo y su usuario (como contactos, fotos, datos GPS) que se cargan en un servidor remoto. También es capaz de cargar toda la tarjeta SD y todos los mensajes SMS almacenados en el dispositivo.	1 4 5 6 10
Steek/Fatakr	Algunas de las muestras tienen la capacidad de robar información privada y enviar mensajes SMS.	3 4 6
TapSnake/Droisnake	Publica la ubicación del teléfono en un servicio web.	5
Tascudap	Esta aplicación se conecta a un servidor remoto y monitoriza los mensajes SMS entrantes. El dispositivo infectado puede ser utilizado para ataques DDoS.	1 6
Tesbo	Establece la conexión a un par de servidores remotos, a los que se remiten detalles del dispositivo como el International Mobile Subscriber Identity (IMSI) y el nombre del paquete de la aplicaciones instaladas.	6
Tetus	Este troyano intercepta todos los mensajes SMS entrantes y los sube a un servidor remoto. También puede eliminar los mensajes SMS en el dispositivo infectado y es capaz de enviar mensajes SMS. Además, el troyano envía una lista de todas las aplicaciones instaladas a un servidor remoto.	4 6
TigerBot	Este malware se comunica con un servidor de C&C a través de mensajes SMS, es capaz de descargar e instalar otras aplicaciones, iniciar las actividades del navegador, actualización instalada binarios, etc...	1 2 6 7
Titan	Este malware ha sido utilizado en ataques dirigidos en Asia y trata de filtrar información sensible. Se propaga a través de mensajes SMS.	1 6
Tonclank	Abre una puerta trasera y descargas archivos en los dispositivos infectados. También roba información del smartphone.	7
TGloader/Stiniter	Escucha a un servidor C&C. Este troyano puede instalar aplicaciones adicionales y enviar mensajes SMS premium.	1 4 7
Tracer	Spyware comercial	1 5 6 8
TypStu	Este troyano roba datos personales.	6
UpdtBot	Servidor de C&C. Las muestras correspondientes tienen la capacidad de instalar paquetes adicionales y enviar mensajes SMS premium.	1 4 7
UpdtKiller	Este troyano detecta y desactiva aplicaciones AV instalados.	8
Uracto	Este malware se usa para engañar a los fans del anime y los jugadores. Roba datos confidenciales.	6
USBcleaver	Cuando el dispositivo está conectado a un equipo con Windows que tiene ejecución automática activada, el troyano intenta reunir información de la computadora, incluyendo: Puerta de enlace predeterminada, Google Chrome contraseña, dirección IP, contraseña de Microsoft Internet Explorer, contraseñas Wi-Fi, etc.	6 10
Uten	Cuando se ejecuta el troyano, informa del estado del dispositivo al atacante y luego descarga un archivo de configuración con las listas de números de teléfono. Después, el troyano envía mensajes SMS a números de teléfono que aparecen en este archivo de configuración. También puede realizar las siguientes acciones adicionales: modificar la configuración del dispositivo, descargar e instalar los nuevos paquetes, intente obtener privilegios de root, etc.	1 2 4 6 7
Uxipp	Este malware intenta enviar mensajes SMS premium.	4
Vdloader	Este malware abre una puerta trasera en el dispositivo infectado y roba datos personales.	2 6
Walkinwat/Pirater	Envía mensajes SMS a todos los números en la guía telefónica y roba información del dispositivo infectado.	4
Waps/Simhosy	Esta aplicación maliciosa intenta robar mensajes SMS y las entradas de contactos en el dispositivo infectado.	6
Wroba/HijackRAT	Esta aplicación maliciosa intenta robar datos privados o credenciales de banca y también sirve como herramienta de acceso remoto (RAT).	1 6 9
YZHC	Este malware envía mensajes SMS premium y bloquea cualquier mensaje entrante que informa al usuario sobre estos servicios. Como otro comportamiento malicioso el malware envía información de privacidad a un servidor remoto.	3 4 6
Zeahache	Abre una puerta trasera y sube la información robada a una URL específica. También envía mensajes SMS.	2 3 4 6
ZergRush	Es una exploit de root para Android 2.2 y 2.3	2
ZertSecurity	Esta aplicacion intenta engañar al usuario para que inserte sus detalles de cuenta bancarios que luego se enviarán a los atacantes.	6 9
Zitmo/Citmo	Trata de robar códigos confidenciales bancarios de autenticación (mensajes mTAN) enviados al dispositivo infectado.	6 9
Zsone	Envía mensajes SMS a números premium.	3 4

LEYENDA:

Actividades	Descripción
1	Funcionalidad de Botnet.
2	Obtiene acceso root o al menos trata de convencer al usuario para rootear su teléfono.
3	Descargable a través de repositorio oficial Google Play.
4	Envía mensajes SMS a sitios de pago o maliciosos.
5	Obtiene información de localización.
6	Envía información robada a servidor/es remoto/s.
7	Instala otras apps o binarios.
8	Potencialmente aplicación no deseada (Herramientas de "Hacker").
9	Troyano bancario, que es capaz de interceptar y modificar los códigos de autenticación de banca (mensajes mTAN).
10	Troyano con la habilidad de infectar un PC con Windows conectado al dispositivo infectado.
11	Cifra todos los datos personales del dispositivo.

FUENTE: http://forensics.spreitzenbarth.de/android-malware/
Última actualización 2 Enero de 2016

ANEXO II - RESULTADOS EVALUACIÓN

Malware	Avast	AVG	Avira	BitDefender
Ackposts	Android:Skymobi-E [PUP]	android/deng	SPR/ANDR.Skymobi.A.Gen	Android.Rsikware.SmsPay.K
Airpush/StopSMS		android/generic	Adware/ANDR.Airpush.L.Gen	
Arspam	Android:Arspam-D [Trj]	android/deng	Android/Agent.20395.29	Android.Trojan.Arspam.A
Basebridge	Android:BaseBridge-AL [PUP]	android/basebridg	Android/BaseBridge.D.Gen	Android.Trojan.BaseBridge.A
Dowgin		android/deng	Adware/ANDR.Dowgin.AJ.Gen	Android.Adware.Dowgin.Z
DroidDream	Android:DrdDream-F [PUP]	android/generic	Android/DroidDream.C.Gen	Android.Trojan.DroidDream.A
DroidKungfu	Android:KungFu-GH [PUP]	android/kungf	Android/DroidKungFu.AT.Gen	Android.Trojan.DroidKungfu.P
FakeDefender	Android:FkDefend-C [Trj]	Sin nombre		Android.Trojan.FakeAV.D
FakeFlash	Android:Fakeapp-CE [Trj]	Android_dc	Android/FakeApp.A.Gen	Android.FakeFlash.G
FakeInst			Android/FakeAV.A.Gen	
FakeTimer	Android:FakeTimer-K [Trj]	android/faketime	Android/FekaTimer.A.Gen	Android.Trojan.FakeTimer.G
Foncy				Android.Trojan.Foncy.A
Geinimi	Android:Geimini-I [Trj]	android/geinim	Android/Geinimi.B.Gen	Android.Trojan.Geinimi.B
HongTouTou/Adrd	Android:Adrd-G [Trj]	android/generic	Android/Adrd.D.Gen	Android.Trojan.AgentSpy.E
Jifake	Android:SMSSend-X [Trj]	Sin nombre	Android/Agent.CJ	Android.Trojan.Jifake.A
Ksapp		android/ksap	Android/MTK.A.Gen	Android.Trojan.Ksapp.C
Locker/SLocker Ransomware	Android:Locker-DQ [Trj]	android/deng	Android/Locker.A.Gen	Android.Trojan.Slocker.F
Obad	Android:Obad-A [Trj]	android/deng	Android/Obad.a.2	Android.Trojan.Obad.A
OpFake	Android:SMSAgent-ACE [PUP]	android/deng	Android/FakeInst.C.1	Android.Riskware.TaJawaBar.A
RootSmart/Bmaster	Android:GGSmart-B [PUP]	android/rootsmar	Android/Agent.CF.Gen	Android.Trojan.RootSmart.A
SMSreg				
Spitmo	Android:Spitmo-M [Trj]	android/spitm	Android/Spy.Spitmo.A.Gen	Android.Trojan.Spitmo.C
TapSnake/Droisnake	Android:TapSnake-A [Trj]	Sin nombre	Android/Tapsnake.A.1	
Tesbo	Android:Tesbo-A [Trj]	Sin nombre	Android/Agent.Q.7	Android.Trojan.Tesbo.A
Titan	ELF:Titan-A [Trj]	Sin nombre	Android/Tediss.A.2	Android.Trojan.Titan.A
UpdtKiller		android/updtkille	Android/UpdtKiller.C	
USBcleaver	Android:UsbCleaver-E [PUP]	android/deng	Android/UsbCleaver.a	Android.Hacktool.UsbCleaver.A
Waps/Simhosy		android/wapz	Adware/ANDR.Waps.I.Gen	Android.Adware.Mobclick.A
ZertSecurity	Android:Zitmo-O [Trj]	android/deng	Android/Spy.Zitmo.B.Gen	Android.Trojan.Zitmo.B
Zitmo/Citmo	Android:Zitmo-A [Trj]	android/zitm	Android/Spy.Zitmo.D.Gen	Android.Trojan.SmsSpy.B
% Detecciones	73,33%	90,00%	90,00%	83,33%

Malware	ESET	Ikarus	Kaspersky
Ackposts	SkyMobi.H	AdWare.AndroidOS.Skymobi	RiskTool.AndroidOS.Skymobi.a
Airpush/StopSMS	AdDisplay.AirPush.S	PUA.AndroidOS.StopSMS	
Arspam	TrojanSMS.Arspam.A	Trojan.AndroidOS.Arsmap	UDS:DangerousObject.Multi.Generic
Basebridge	BaseBridge.P	Exploit.AndroidOS.DroidRooter	UDS:DangerousObject.Multi.Generic
Dowgin	AdDisplay.Dowgin.R	AdWare.AndroidOS.Umeng	AdWare.AndroidOS.Dowgin.d
DroidDream	Lightdd.C	Trojan.AndroidOS.DroidDream	UDS:DangerousObject.Multi.Generic
DroidKungfu	DroidKungfu.D	Trojan.AndroidOS.DroidKungFu	UDS:DangerousObject.Multi.Generic
FakeDefender	FakeAV.C		UDS:DangerousObject.Multi.Generic
FakeFlash	FakeApp.D	Trojan.AndroidOS.FakeApp	UDS:DangerousObject.Multi.Generic
FakeInst		Trojan.AndroidOS.FakeInst	
FakeTimer	FakeTimer.D	Trojan.AndroidOS.FakeTimer	UDS:DangerousObject.Multi.Generic
Foncy			
Geinimi	Spy.Geinimi.A	Trojan.AndroidOS.Geinimi	UDS:DangerousObject.Multi.Generic
HongTouTou/Adrd	Adrd.A	Trojan.AndroidOS.Adrd	UDS:DangerousObject.Multi.Generic
Jifake	TrojanSMS.Agent.E	Trojan.AndroidOS.Jifake	UDS:DangerousObject.Multi.Generic
Ksapp	AdDisplay.Waps.H	Android.Ksapp	UDS:DangerousObject.Multi.Generic
Locker/SLocker Ransomware	Locker.B	Trojan-Ransom.AndroidOS.FBILocker	UDS:DangerousObject.Multi.Generic
Obad	Obad.B	Backdoor.AndroidOS.Obad	UDS:DangerousObject.Multi.Generic
OpFake	TrojanSMS.Agent.F	Trojan-SMS.AndroidOS.Opfake	UDS:DangerousObject.Multi.Generic
RootSmart/Bmaster	GGSmart.A	Trojan.AndroidOS.GGSmart	UDS:DangerousObject.Multi.Generic
SMSreg		Android.SMSreg	
Spitmo	Spy.Spitmo.A	Trojan-Spy.AndroidOS.Spitmo	UDS:DangerousObject.Multi.Generic
TapSnake/Droisnake	Spy.TapSnake.A	Trojan.AndroidOS.GPSSpy	UDS:DangerousObject.Multi.Generic
Tesbo	TrojanSMS.Agent.AXT	Trojan.AndroidOS.Tesbo	UDS:DangerousObject.Multi.Generic
Titan	Raidum.A	Android.Tediss	UDS:DangerousObject.Multi.Generic
UpdtKiller	UpdtKiller.B	Trojan.AndroidOS.UpdateKiller	UDS:DangerousObject.Multi.Generic
USBcleaver	UsbCleaver.A	Trojan.AndroidOS.USBCleaver	UDS:DangerousObject.Multi.Generic
Waps/Simhosy	AdDisplay.Waps.L	PUA.AndroidOS.Waps	
ZertSecurity	Spy.Zitmo.B	Trojan.AndroidOS.Zitmo	UDS:DangerousObject.Multi.Generic
Zitmo/Citmo	Spy.SmsSpy.B	Trojan.AndroidOS.Zitmo	UDS:DangerousObject.Multi.Generic
% Detecciones	90,00%	93,33%	83,33%

Malware	Intel Security	Sophos	Trend Micro
Ackposts	Android/PUP.Skymobi.C!Gen	Android Skymobi Pay	PUA:AndroidOS_Skymobi.OPS
Airpush/StopSMS	Adware.AirPush.A!Gen	Android Airpush	
Arspam	Android/Arspam.A!apk	Andr/Arspam-A	AndroidOS_Arspam.HAT
Basebridge	Android/G.1313!dex	Andr/DroidRt-A	AndroidOS_BasBrideg.HAT
Dowgin		Android Dowgin	
DroidDream	Android/DrdDream.N!apk	Andr/RootCage-A	AndroidOS_DroidDream.CT
DroidKungfu	Android/DroidKungfu.E!apk	Andr/KongFu-B	AndroidOS_Kungfu.HATA
FakeDefender	Android/Fakedefender.C!Gen	Andr/FkDefend-A	AndroidOS_FakeAv.HAT
FakeFlash	Android/Fladstep.B!Gen	Andr/FakeApp-F	AndroidOS_FakeApp.HATC
FakeInst			
FakeTimer	Android/OneClickFraud.G3!Gen	Andr/FkTime-D	AndroidOS_FAKETIMER.CLT
Foncy		Andr/RuFraud-B	
Geinimi	Android/Geinimi.A!apk	Andr/Geinimi-I	AndroidOS_Geinimi.HRXA
HongTouTou/Adrd	Android/Drad.B1!apk	Andr/Geinimi-B	PUA:AndroidOS_Adrd.HRX
Jifake	Android/Jifake.L!apk	Andr/Jifake-E	AndroidOS_Jifake.HQT
Ksapp	Android/GSD.A!Gen	Andr/Ksapp-A	AndroidOS_Ksapp.HQTA
Locker/SLocker Ransomware	Android/ScarePackage.A!Gen	Andr/Slocker-AA	AndroidOS_Ransom.CLT
Obad	Android/Obad.A!Gen	Andr/Obad-A	AndroidOS_Obad.HAT
OpFake	Android/G.1910!dex	Andr/FakeMini-A	AndroidOS_Fakebrows.HAT
RootSmart/Bmaster	Android/RootSmart.A!apk	Andr/GGSmart-A	AndroidOS_Rooter.HNT
SMSreg			
Spitmo	Android/Spitmo.B!apk	Andr/Spitmo-A	AndroidOS_Siptmo.HRX
TapSnake/Droisnake	Android/Spyware.GpsNake.A1!dex		AndroidOS_Snakspy.HQT
Tesbo	Android/G.46c7c8	Andr/SMSRep-Z	AndroidOS_Tesbo.HNTA
Titan	Android/Tediss.A!exe	Andr/Tediss-B	AndroidOS_TitanTrj.ISE
UpdtKiller	Android/UpdtKiller.B!Gen		AndroidOS_Jnyl.HBT
USBcleaver		Android USB Cleaver	AndroidOS_UsbCleaver.HQT
Waps/Simhosy		Android Wapsx	
ZertSecurity	Android/Zitmo.I!Gen	Andr/Zitmo-D	AndroidOS_Zitmo.HBTA
Zitmo/Citmo	Android/Zitmo.E!apk	Andr/SMSRep-B	AndroidOS_Spitmo.CLT
% Detecciones	80,00%	86,67%	80,00%