Malware
Ensamblador y Mobile

Manuel García-Cervigón

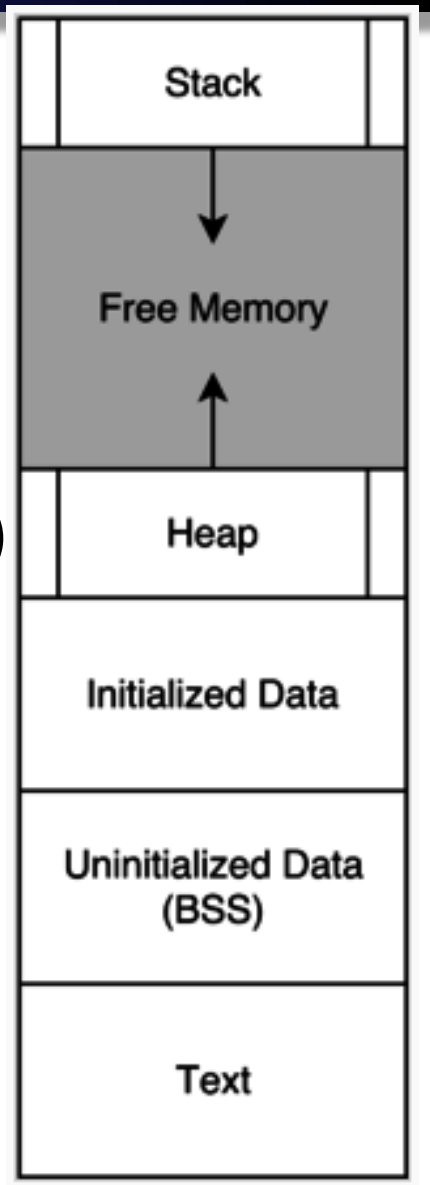
- Parte de los contenidos de esta presentación se basan en el libro “Hacking, Técnicas Fundamentales”

Lenguaje ensamblador

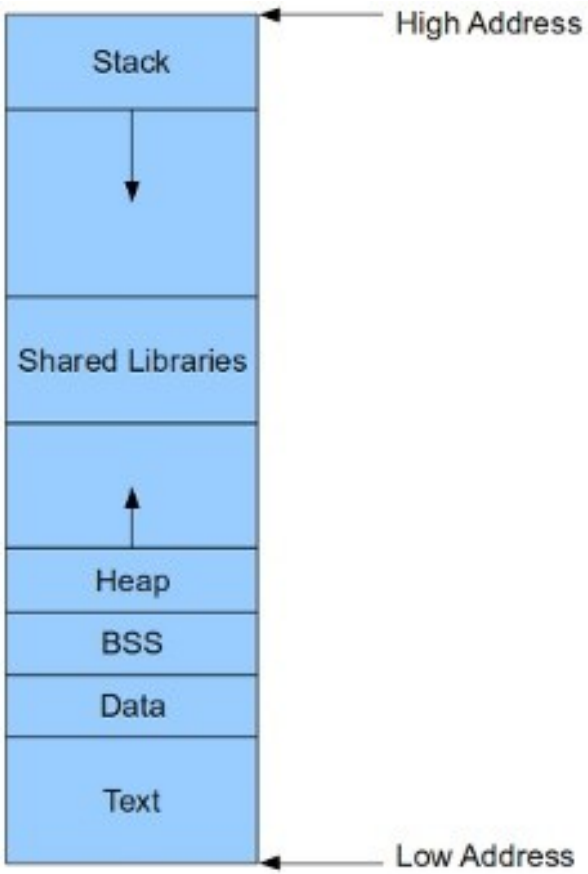
- Cada vez que se ejecuta cualquier programa, el mismo deberá pasar a memoria.
- Los programas en memoria tienen varias secciones o *segmentos , los cuales sirven para organizar el manejo de la memoria por el programa.*

Lenguaje ensamblador

- Los Segmentos de Memoria son:
  - CS. Code Segment (Segmento de Código)
  - DS. Data Segment (Segmento de Datos)
    - HS. Heap Segment (Segmento de Heap, Dynamic memory allocation)
    - BSS (uninitialized data, variables & constants)
    - Data (global, static)
  - SS. Stack Segment (Segmento de Pila)
- Cada Segmento tiene asignado una "cosa" del programa



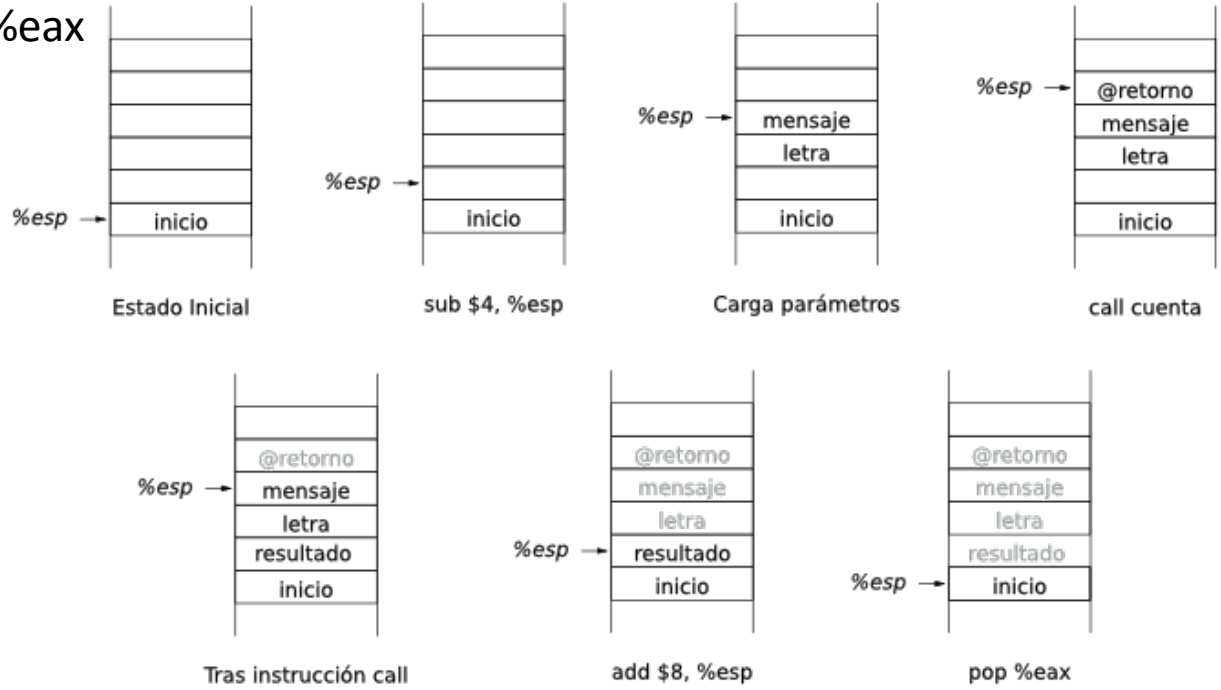
Lenguaje ensamblador



Ensamblador
Segmento de Pila

- En este segmento se guardan:
  - Los parámetros de las funciones llamadas
  - Las variables locales
  - Otra información necesaria para el funcionamiento del programa.
- Cada vez que se llama a una función entra en este segmento con toda su información.
- Ejemplo:
  - http://ocw.uc3m.es/ingenieria-telematica/arquitectura-de-ordenadores/lecturas/html/sub.html#sub:exa:exasubcode

1. ...
2. sub $4, %esp # Espacio para el resultado
3. push letra # Parámetros en el orden correcto
4. push mensaje
5. call cuenta # Invocación de la subrutina
6. add $8, %esp # Descarga del espacio para parámetros
7. pop %eax # Resultado en %eax
8. ...



Desbordamiento de pila

- Cuando se llama a una función recursiva, cada llamada que se haga irá entrando al segmento de pila.
- Si la recursión es demasiado “profunda”, corremos el riesgo de llenar la pila, y *revasarla*

Desbordamiento de heap
Ejemplo:Notetacker

- La memoria del montón (heap) se asigna mediante el comando malloc()
- Aplicación que toma notas y las guarda junto con el id del usuario en /var/notes

  - buffer = (char *) ec_malloc(100);
  - datafile = (char *) ec_malloc(20);
  - strcpy(datafile, "/var/notes");

Desbordamiento de heap
Ejemplo:Notetacker.c

- ./notetaker $(perl –e ‘print “A”x104’)
- ./notetaker $(perl –e ‘print “A”x104 . “tess ile”’)

- Podríamos aprovecharlo para sobreescribir…/etc/passwd

- perl –e ‘print “myroot:XXq2wKiyI43A2:0:0” . “A”x68 . “:/root:/tmp” . “/etc/passwd”’

Malware analysis. Behavioral analysis

1. Fingerprint: md5sum
2. Antivirus: www.virustotal.com
3. Registry changes: Regmon
4. System info: Process Monitor
5. Capture traffic: CaptureBat/wireshark
6. Simulate servers: Mailpor, Fake DNS

- "A disassembler will take a binary and break it down into human readable assembly."(static analysis)
- With a debugger we can step through, break and edit the assembly while it is executing (dynamic analysis).

Malware analysis. Behavioral analysis

Interesting tools for preliminar forensic analysis:

http://download.sysinternals.com/Files/SysinternalsSuite.zip

Filemon.exe: Allows us to "strace -eopen,read,write..."
Regmon.exe: Monitors windows registry accesses
Tcpview.exe: "netstat -tunl" equivalent, can kill connections
Procexp.exe: Allows us to attach to a process using windbg
WinObj.exe: Displays NT objects.

Malware analysis. Code analysis

1. Fingerprint: md5sum
2. Antivirus: www.virustotal.com
3. Details about the PE (Portable Executable (PE) format): PEiD.
    1. Is it Packed?
4. Readable code: strings
5. More Details about the PE: PEview
    1. Imports
    2. Exports
    3. Metadata
    4. Resources
6. Disassembly

Lenguaje ensamblador

- **GDB**
  - gdb –q
  - set dis intel
  - quit

>type hello.c

```
#include <stdio.h>
main() { prin;("just hello");
 return 0; }
```

>gcc -gstabs hello.c –ohello
>gdb –qhello

(gdb) run
StarDng program: /RAM Disk/hello just hello Program terminated with signal SIGQUIT, Quit. The program no longer exists.
(gdb) quit

Lenguaje ensamblador
Ejemplo x86-64

- rax <= 64bits
- eax <= 32bits
- ax <= 16bits
- al / ah <= 8 bits

- RBP: Base pointer (EBP en 32 bits)
- RSP: Stack pointer. Direccion del primer elemento de la pila (ESP en 32 bits)
- RIP: Instruction pointer (EIP en 32 bits)

Firstprog.c

```
#include <stdio.h>
int main()
{
int i;
      for (i=0;i<10;i++)
      {
      puts("Hello, world\n");
      }
      return 0;
}
```

Mobile Malware

Introducción

- “El 85% de la población tiene un dispositivo móvil”
- “El 90% toma fotos con el celular”

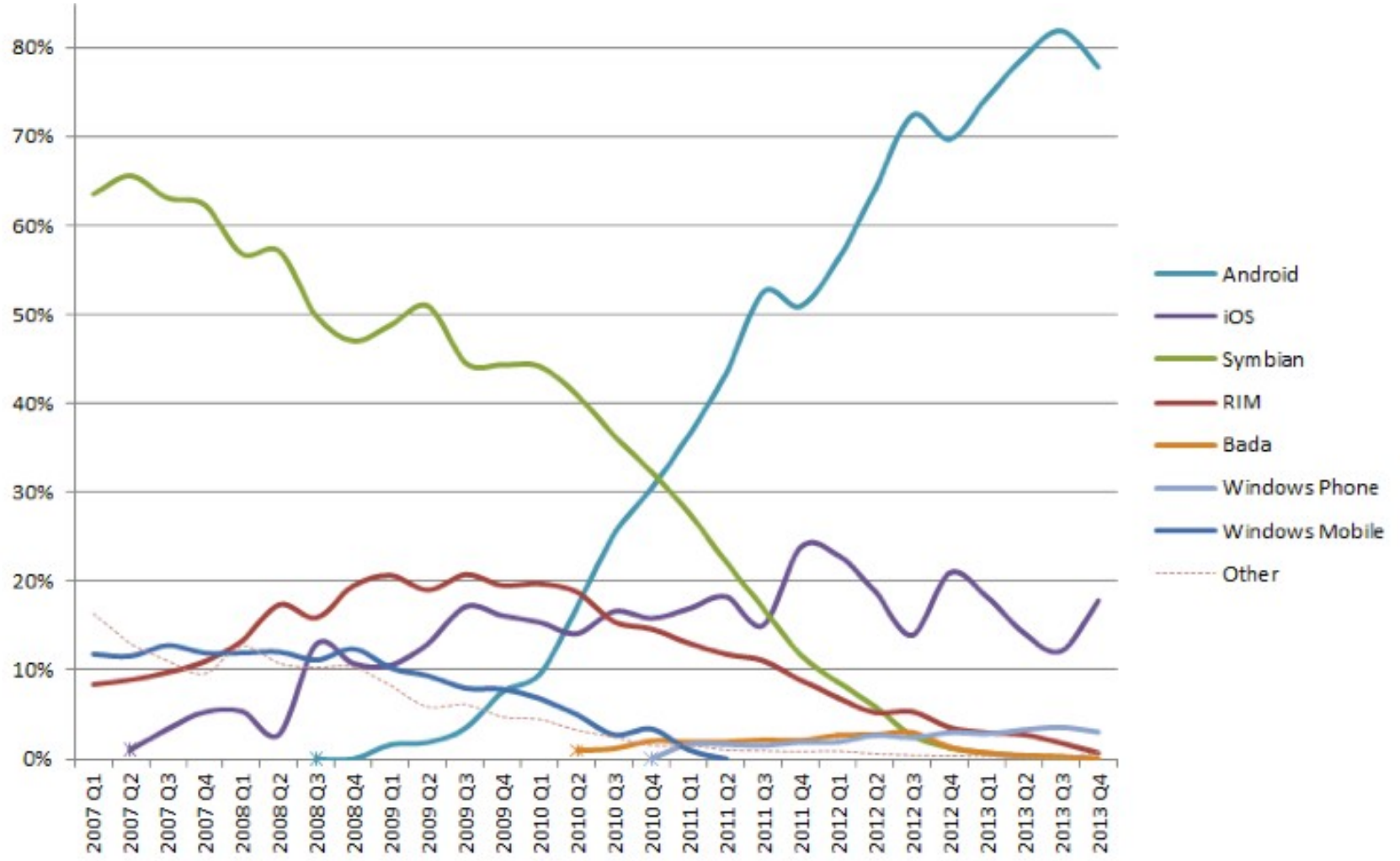
- “El 98% del malware se enfoca a Android”

  Computer World 2014

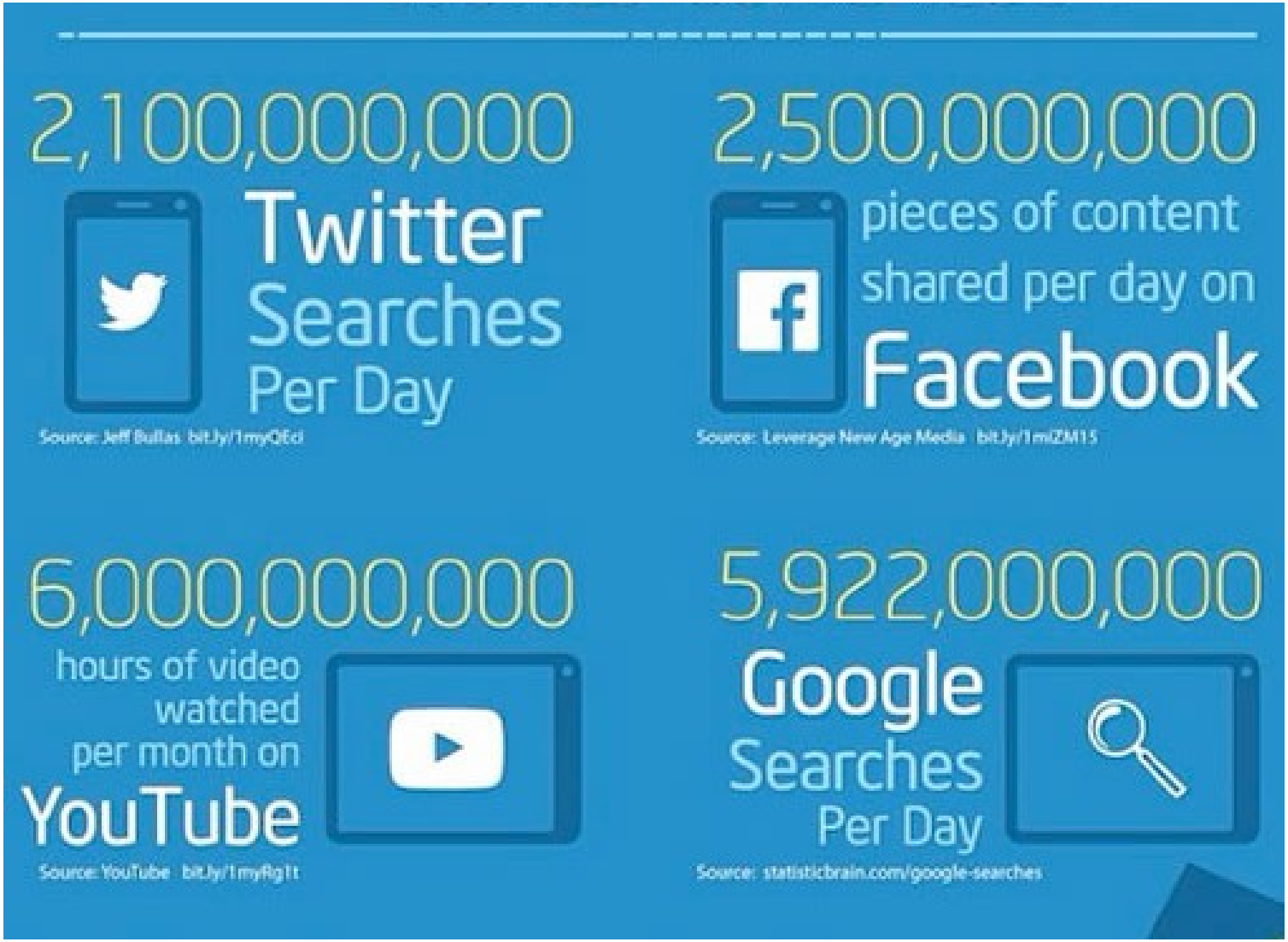
- Los celulares han aumentado su pontencia de forma exponencial
- El 60% del malware forma parte de un botnet
- Algunos malware tienen técnicas antiborrado (Svpeng)

- http://www.statista.com/statistics/325159/malicious-mobile-programs-2014/

Introducción



Introducción

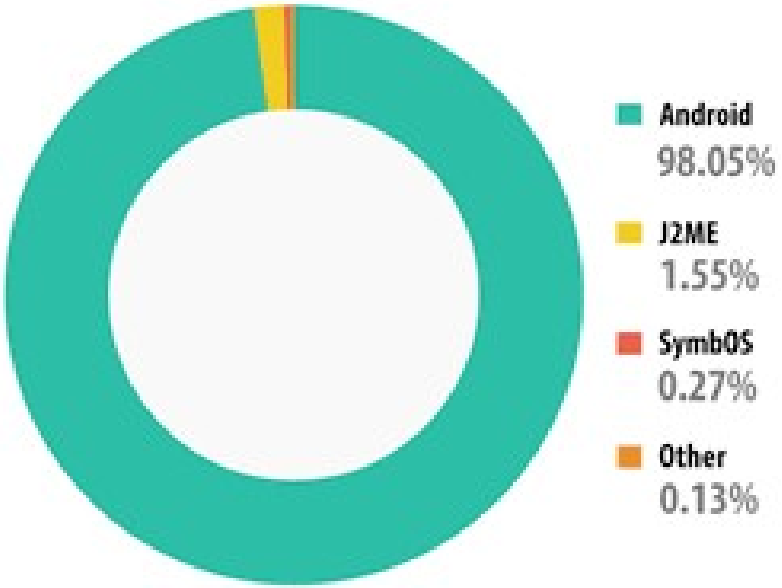


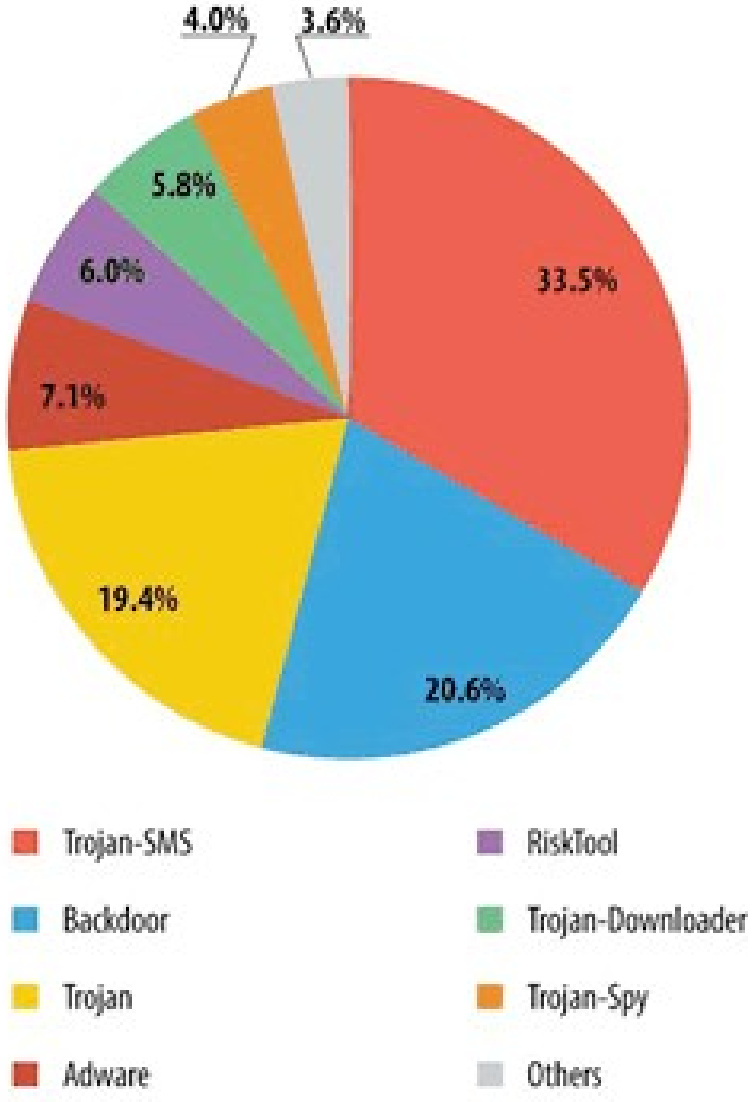
Phishing trends report
3Q2014

The majority of malware threats do not belong to new families developed from scratch, but are variants of well-known malware specimens modified by their creators to evade detection systems

Malware Infections by Type	% of malware samples
Trojans	75.00%
Viruses	1.47%
Worms	2.09%
Adware/Spyware	6.88%
Other	14.55%

Introducción

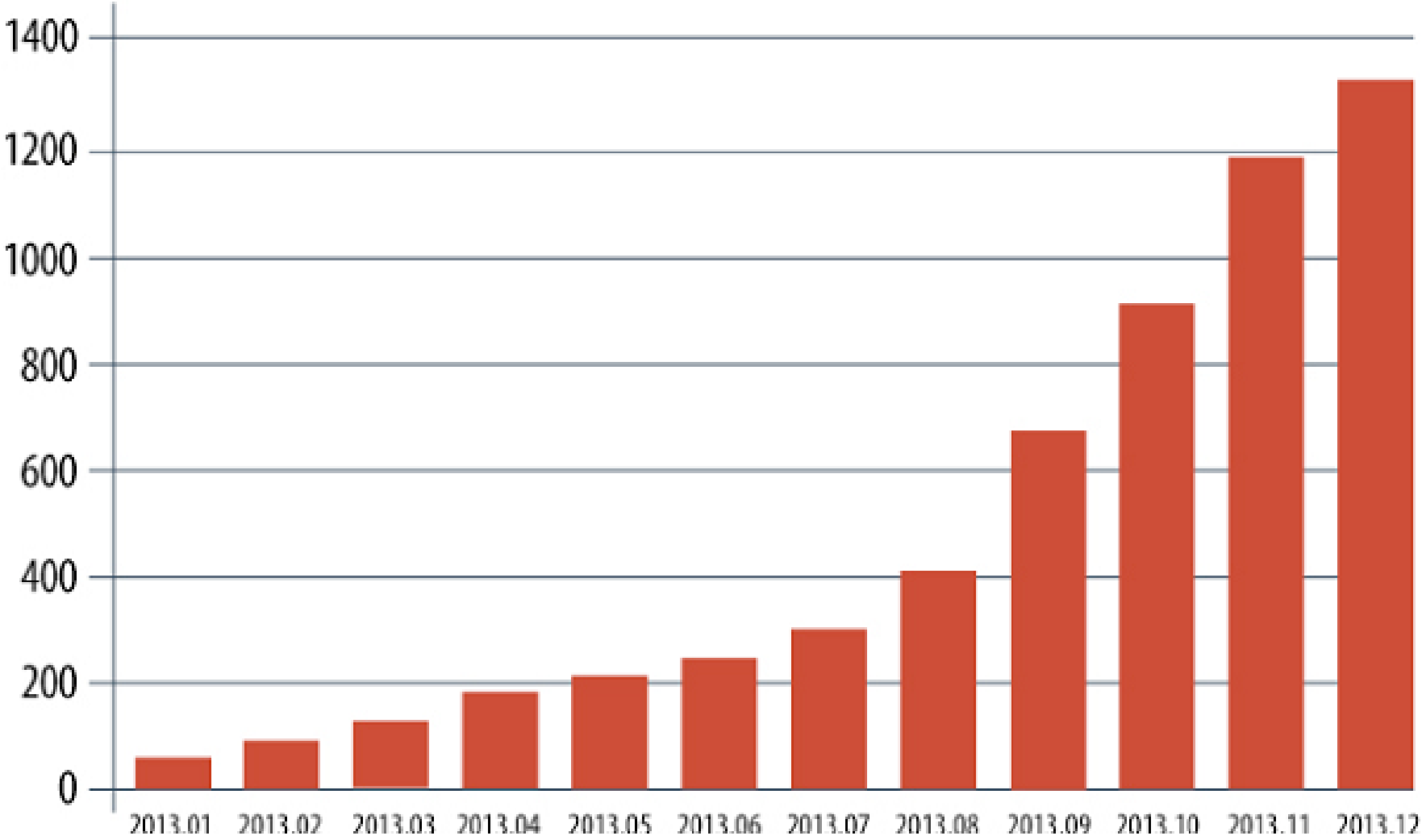




Kaspersky Labs:

Mobile malware
detected in 2013
by platform and category

Introducción Evolución de troyanos



Introducción

- Extravío del teminal
- Robo
- Malware
- Fraude telefónico:
  - La recepción de mensajes cortos de texto que ofrecen servicios que el usuario no ha requerido
  - SMS pidiendo que se visite una página web sospechosa de ser fraudulenta
  - solicitud de claves de usuario o información personal

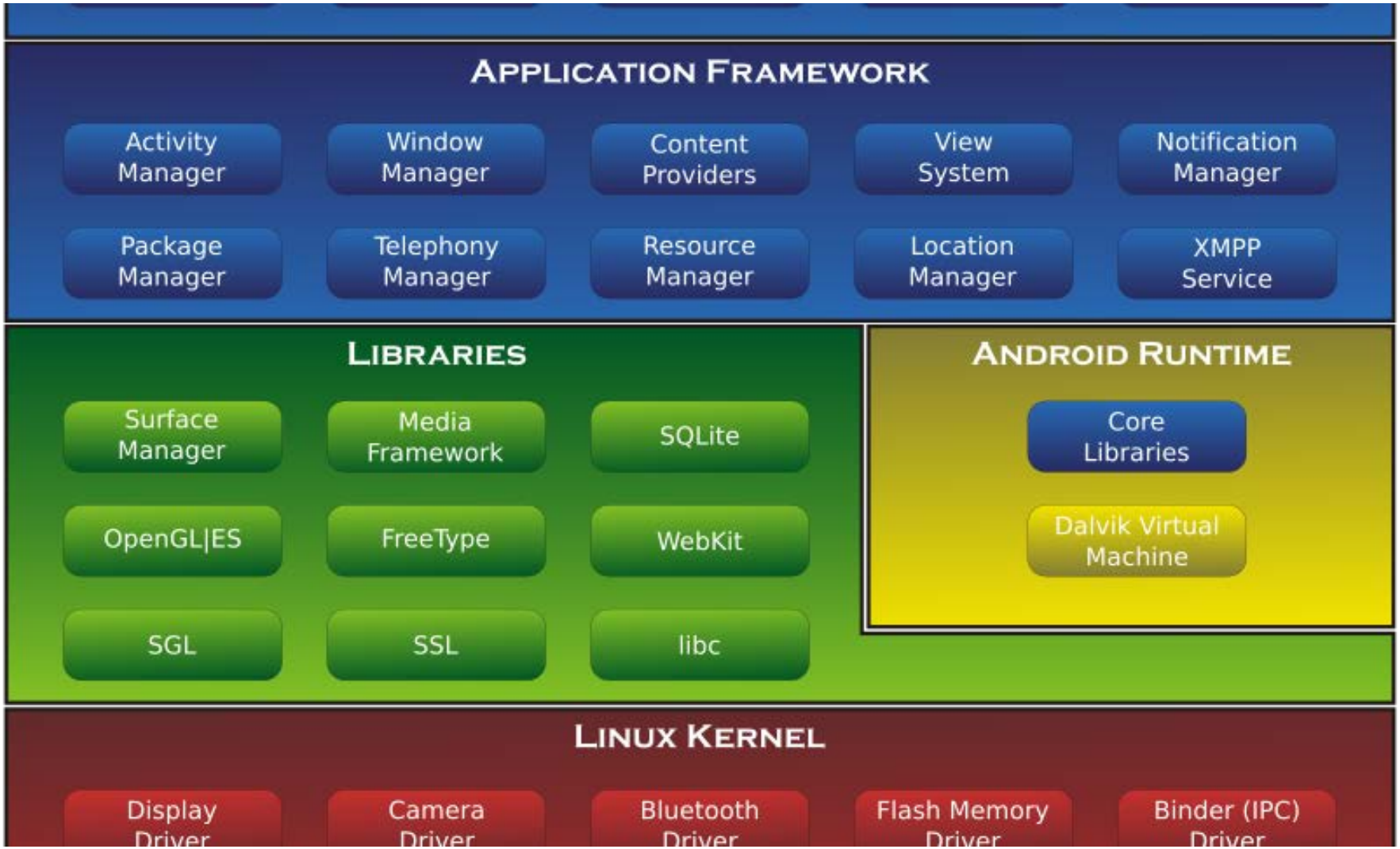
Seguridad en android

- El número de malware en sistemas móviles ha aumentado de manera exponencial en los últimos años.
- La implantación de antivirus también ha crecido (16%)
- Uso generalizado de PIN (84%)
- Igualmente aumenta el número de personas que activan el bloqueo del móvil por desuso (21%)

Seguridad en Android

- Arquitectura en capas
- Sandbox
- Usuario root
- Modelo de permisos

Seguridad en Android
Arquitectura en capas



Seguridad en Android
Sandbox

- Cada aplicación se ejecuta en un Sandbox propio
- Cada aplicación tiene un usuario único
- Políticas multiusuario único
- Políticas multiusuario de Linux
- Protección a nivel de kernel

Seguridad en Android
Usuario root

No rootear:

- Solo un pequeño grupo de servicios tienen permisos de root
- El usuario no tiene acceso a root

Rootear:

- Acceso il.limitado al sistema Rootear
- un movil elimina su seguridad

A favor:

- Sistema sencillo
- Protegen los recursos
- Son infalibles

En contra:

- El usuario acostumbra a aceptar todos!

Seguridad en Android
Seguridad externa

- Detección a nivel de móvil (anMvirus)
- AnMvirus de nivel de Market AnMvirus
- a nivel de Red

CONAN mobile

¿Qué es CONAN mobile?

Una aplicación GRATUITA que realiza la comprobación integral de SEGURIDAD de tu smartphone y tableta.

Estado global de seguridad de tu dispositivo.

Análisis de seguridad detallado:

Configuración

Análisis de la configuración del sistema operativo Android que pueden suponer un riesgo de seguridad.

Propiedades de configuración
Informa si hay configuradas opciones no seguras en el dispositivo.

Redes Wi-Fi inseguras
Listado de las redes WiFi cuya configuración es insegura.

Dispositivos Bluetooth vinculados
Información de los dispositivos emparejados al smartphone o tablet.

Seguridad en Android
Seguridad externa

Aplicaciones

Análisis de peligrosidad de las aplicaciones instaladas en el dispositivo. Los resultados del análisis pueden ser:

Maliciosa: listado de aplicaciones que han sido falsificadas o que tienen un comportamiento peligroso según varios antivirus.

Sospechosa: es detectada como peligrosa por varios antivirus.

Permisos

Clasificación de los permisos que declaran las aplicaciones en función de la peligrosidad de los mismos, de acuerdo con el riesgo para la seguridad establecido por Google.

Alto **Medio** **Bajo** **Otros**

Servicio proactivo

Alerta a los usuarios de comportamientos anómalos y potencialmente maliciosos.

Eventos

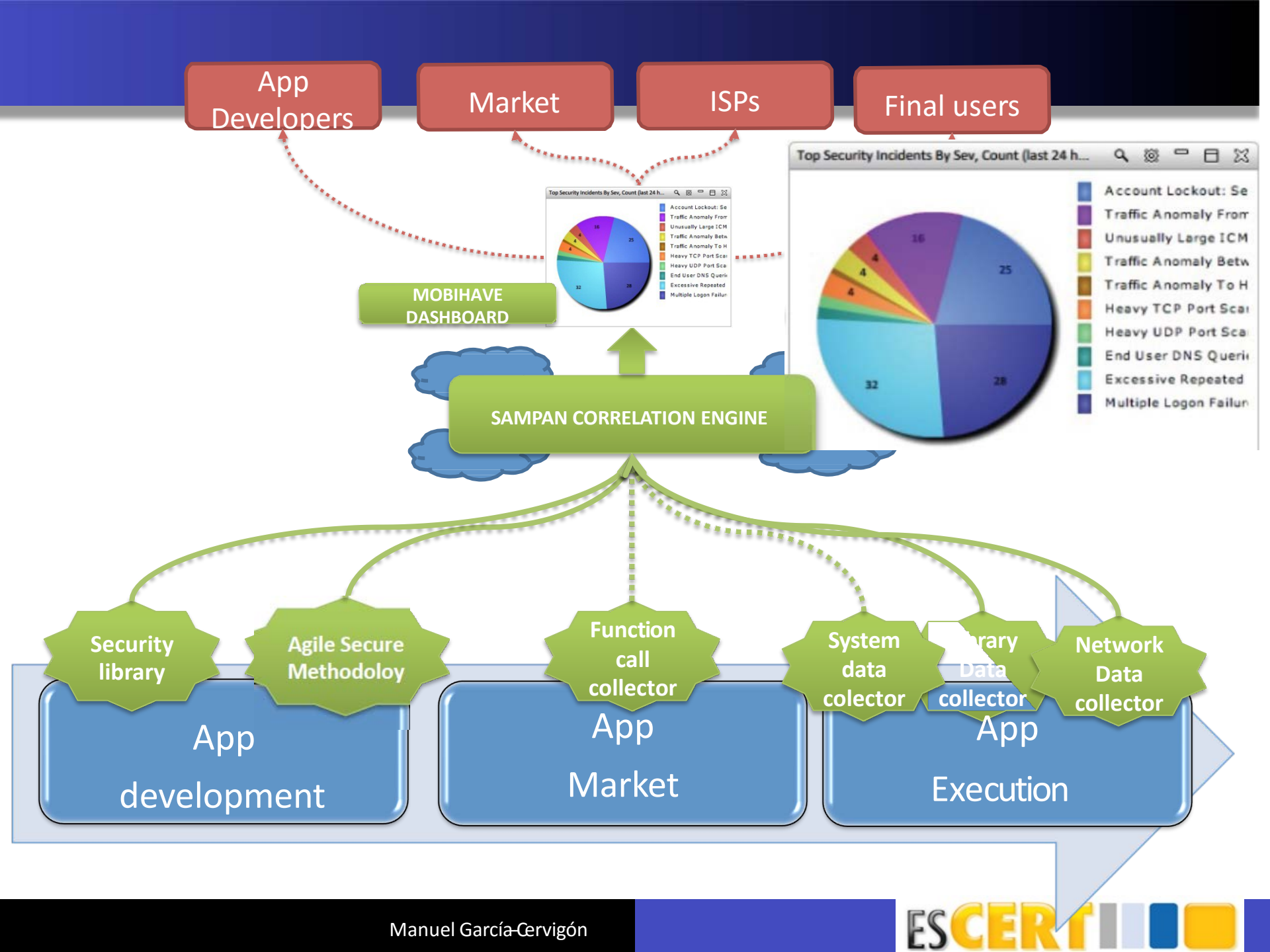
- Cambios en el fichero /etc/host
- Detección de conexiones a redes WiFi inseguras
- Comprobación de nuevas aplicaciones
- Llamadas y envíos de mensajes a números de tarificación especial (servicios Premium)
- Detección de conexiones potencialmente maliciosas
- Detección de amenazas de seguridad relacionadas con Botnets

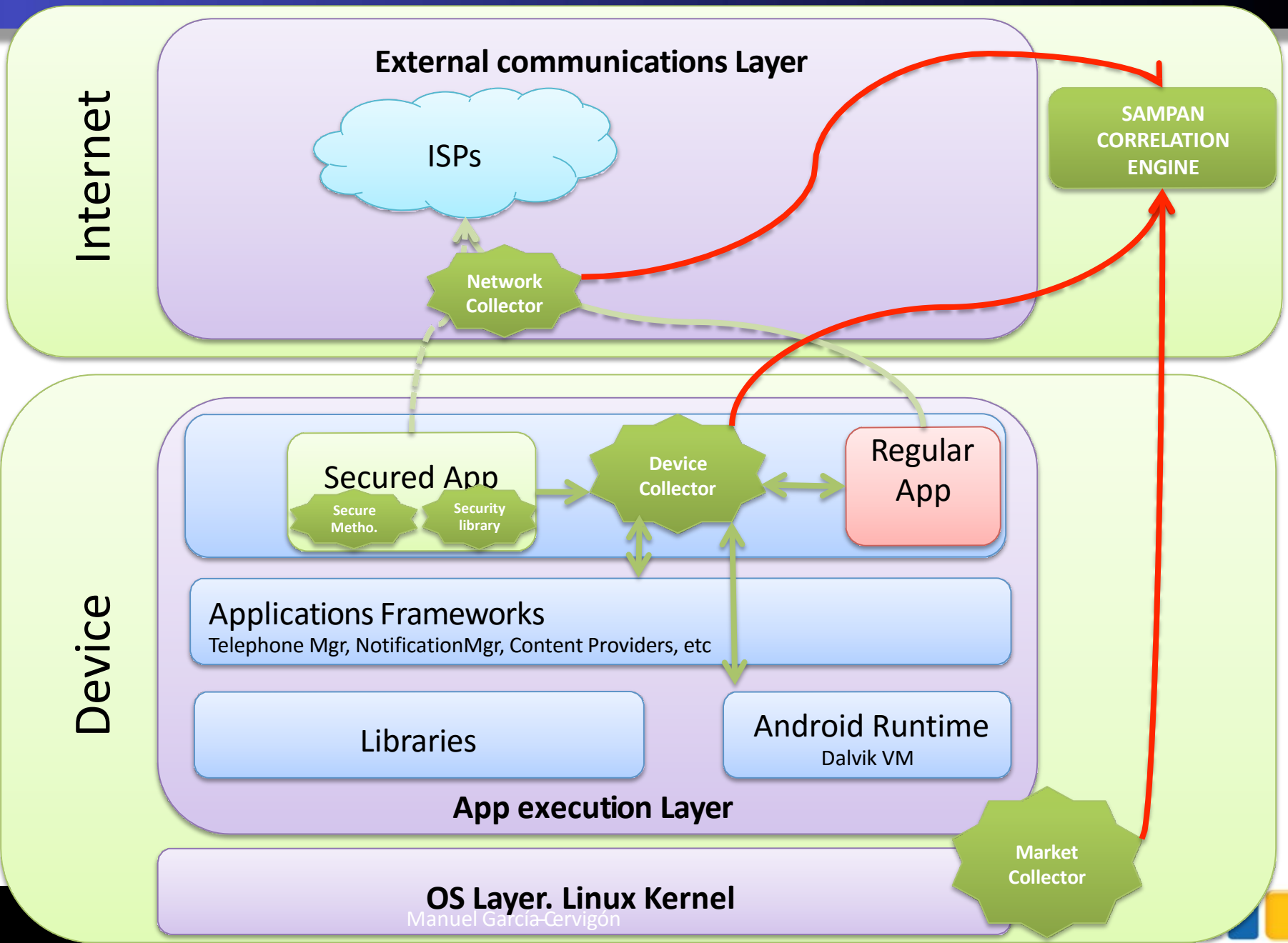
Conexiones

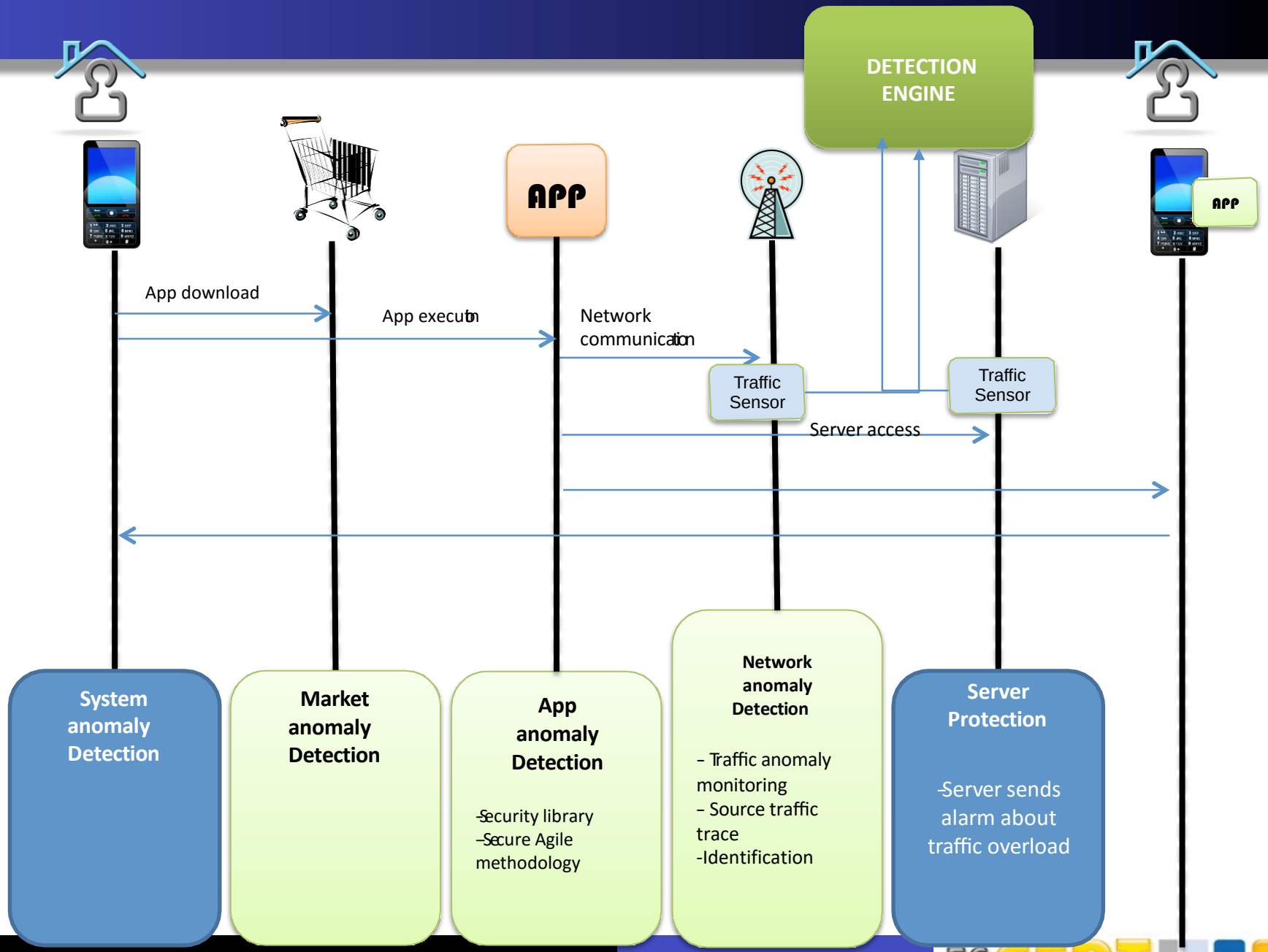
Muestra las conexiones de red realizadas por las aplicaciones instaladas, advirtiendo al usuario de aquellas que son potencialmente maliciosas.

Además se podrá obtener información extendida de cada una de las conexiones, como por ejemplo, la geolocalización de la dirección IP.

Análisis de aplicaciones


App Developers
Market
ISPs
Final users
MOBIHAVE DASHBOARD
SAMPAN CORRELATION ENGINE
Top Security Incidents By Sev, Count (last 24 h...
Account Lockout: Se
Traffic Anomaly From
Unusually Large ICM
Traffic Anomaly Betw
Traffic Anomaly To H
Heavy TCP Port Scar
Heavy UDP Port Sca
End User DNS Queri
Excessive Repeated
Multiple Logon Failur
Security library
Agile Secure Methodoloy
Function call collector
System data colector
Network Data collector
App development
App Market
App Execution


Internet
External communications Layer
ISPs
Network Collector
SAMPAN CORRELATION ENGINE
Device
Secured App
Secure Metho.
Security library
Device Collector
Regular App
Applications Frameworks
Telephone Mgr, NotificationMgr, Content Providers, etc
Libraries
Android Runtime
Dalvik VM
App execution Layer
Market Collector
OS Layer. Linux Kernel


DETECTION ENGINE
APP
APP
App download
App execution
Network communication
Traffic Sensor
Traffic Sensor
Server access
System anomaly Detection
Market anomaly Detection
App anomaly Detection
-Security library
-Secure Agile methodology
Network anomaly Detection
- Traffic anomaly monitoring
- Source traffic trace
-Identification
Server Protection
-Server sends alarm about traffic overload

Análisis de aplicaciones
Estado del arte (academia)

	Llibreries o syscalls	Rendiment	Eficàcia	Root?
CrowDroid	syscalls	Bo	?	Sí
Paranoid	syscalls	Bo	Depèn	Sí
Aurasium	llibreries	Regular	Alta	No

Table: Comparació CroiwDroid, Paranoid Android, Aurasium

Resultados poco contrastados
Bajo rendimiento
Consumo de batería
Reempaquetado (aurasium)
Poco escalable

Análisis de aplicaciones Busquemos soluciones

EINA	TIME (ms)	PSS (MB)	CPU (ns)	SORTIDA (KB)
Cap	14.111,78	7,86	10.092.579.077	0,00
AM	4,54%	102,26%	10,02%	8.171,66
PIN PROCCOUNT	116,77%	928,61%	159,07%	66,70
PIN PROCTRACE	*disc overflow*			
PIN PROCTRACE GZIP	*ANR*			
PIN SYSCALLCOUNT	98,78%	158,84%	127,10%	0,24
PIN SYSCALLTRACE	96,07%	161,14%	129,25%	369,94
PIN SYSCALLTRACE GZIP	97,03%	163,45%	129,67%	17,77
STRACE COUNT	0,49%	0,11%	0,06%	2,64
STRACE TRACE	1,68%	1,10%	0,57%	428,95

Objetivo

Un APK fácil de distribuir que instrumente las llamadas, eficiente y que proporcione modelos de normalidad