DECSAI
Departamento de Ciencias de la Computación e I.A.
Universidad de Granada

Seguridad en dispositivos móviles

© Fernando Berzal, berzal@acm.org

Seguridad en dispositivos móviles

- Dispositivos móviles
  - Seguridad y privacidad
  - Casos de uso
- Seguridad en Android
  - Problemas de configuración
  - Rooting / jailbreaking
  - El ecosistema Android
  - Androidismos
  - Herramientas
  - Vulnerabilidades más habituales
- Caso práctico: WhatsApp
- IoT: coches [autónomos], drones, domótica, fitness…

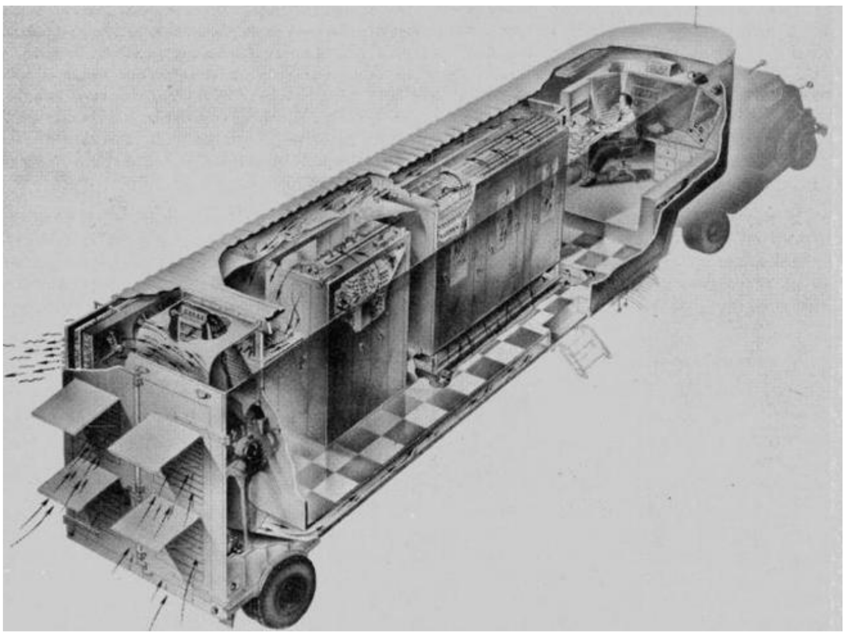
Dispositivos móviles

Disco duro “portátil” de IBM, 5MB en 1956…

Dispositivos móviles

DYSEAC ¿el primer ordenador portátil?

National Bureau of Standards, 1954

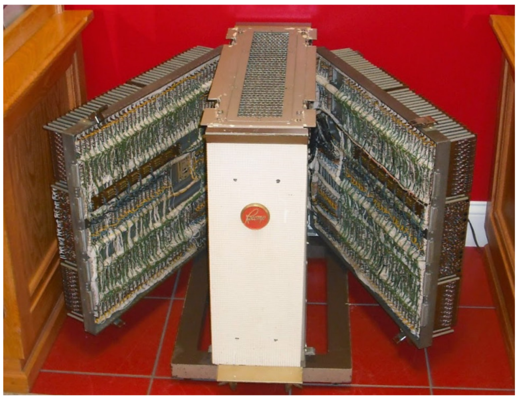


12 metros
900 válvulas
24500 diodos
512x45 bits

Dispositivos móviles

RECOMP 501 [Reliable COMPuter]

North American Aviation, 1958



RECOMP

a 200-lb. portable digital computer that can solve your problems where they happen

It may be a highway construction job . . . an oil exploration . . . an aerial survey. Wherever your problems happen—in the field, office, or lab—Recomp can solve them on the spot. Recomp's all-transistor design makes it the first general-purpose digital computer that's so lightweight, compact, rugged it can travel anywhere.

But Recomp is no lightweight when it comes to performance. It solves the same kind of scientific analysis, data reduction, and business problems that the gigantic vacuum-tube machines solve. And it can out-perform any computer in its price range.

For example, in just 5 minutes Recomp can solve bridge design problems that would take 40 man-hours by standard methods. Recomp can also figure cut and fill, borrow-pit, traffic analysis and simulation, roadway design, grade profile, and drainage problems in just a fraction of the time formerly required. The construction bid itself can be prepared faster and more accurately than ever. And even though Recomp solves the most complex problems, it is remarkably simple to operate. Semi-skilled personnel can learn how in a few hours.

Recomp provides a level of reliability unobtainable in any other computer. Its transistorized etched circuits are mounted on plug-in cards. There are only a few types of cards, and they are easy to replace if servicing is required. Service is available throughout the U.S.

Recomp doesn't require a special air-conditioned room. Plug it into any 115-volt outlet . . . power it with a field generator. It uses less current than a toaster.

For complete details, prices, and brochure, write Marketing Director, Autonetics, Dept. R-12, 9150 E. Imperial Highway, Downey, California.

Oil Exploration. Recomp can compute blast data to figure depth, location, and size of rock and mineral formations in a fraction of the time required by standard methods.

Research and Development. In the lab, Recomp quickly solves complex R&D problems. It takes up no more room than a desk. Just plug it in and it's ready to go to work.

NERVE CENTER OF THE NEW INDUSTRIAL ERA **Autonetics**
A DIVISION OF NORTH AMERICAN AVIATION, INC.

200 libras (90 kg)

Transistores

Memoria de tambor

Cintas perforadas

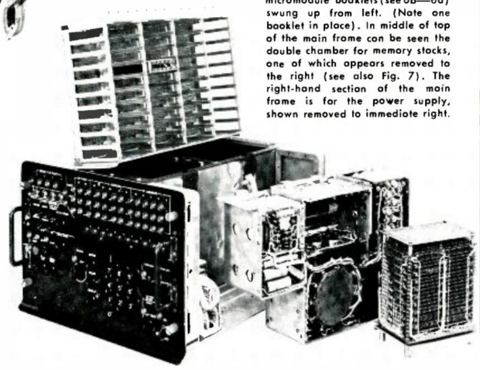
Dispositivos móviles

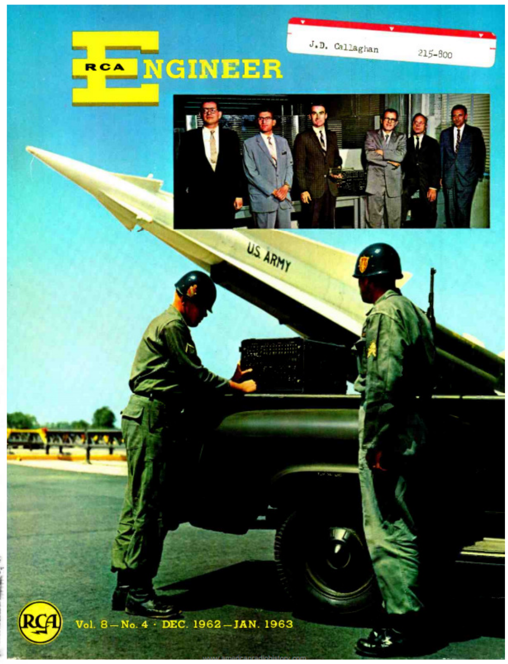
MICROPAC

1962

Fig. 1—MICROPAC case and controls.

Fig. 6a—Disassembled MICROPAC. Center: main frame, with rack for micromodule booklets (see 6b—6d) swung up from left. (Note one booklet in place). In middle of top of the main frame can be seen the double chamber for memory stocks, one of which appears removed to the right (see also Fig. 7). The right-hand section of the main frame is for the power supply, shown removed to immediate right.





50 kg

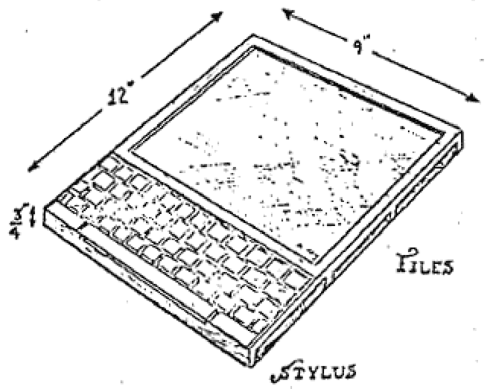
a.k.a. breakie-backie [“rompe-espaldas”]

Dispositivos móviles

Dynabook

"A personal computer for children of all ages"

Alan Kay, Xerox Palo Alto Research Center, 1972

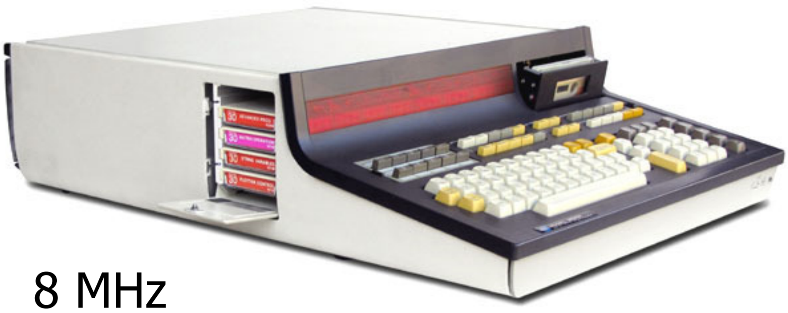


"A standalone 'smart terminal' that uses one of these chips for a processor (and includes memory, a keyboard, a display and two cassettes) is now on the market for about $6000."

Dispositivos móviles

HP-9830A calculator

Hewlett-Packard, December 1972



8 MHz

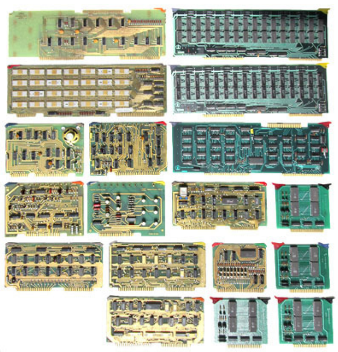
4KB / 8KB RAM

LED 32 caracteres

Cassette

45 lb (20 kg)

$ 5 975

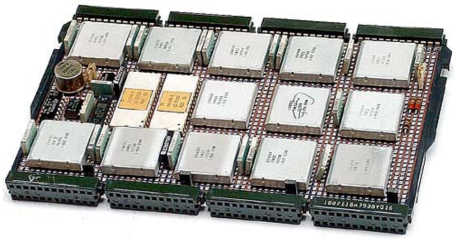


Dispositivos móviles

IBM 5100 Portable PC

IBM, September 1975

1.9 MHz
64KB RAM
5″ monocromo
Cinta DC300
55 lb (25 kg)
$ 19 975

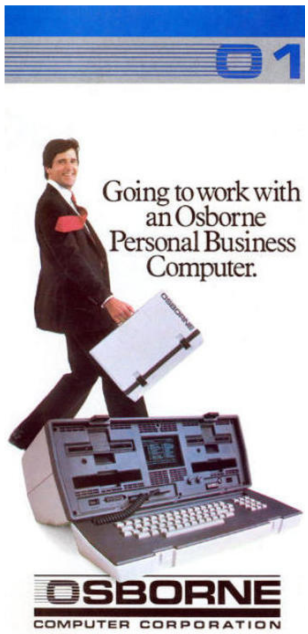


CPU (>15 chips)

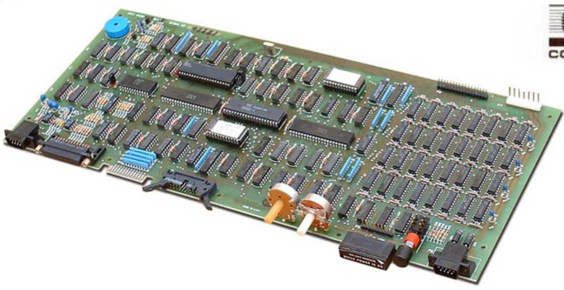
Dispositivos móviles

Osborne 1

Osborne Computer Corporation, April 1981



Zilog Z80 4MHz
64 KB RAM
5″ CRT
Discos 5¼″
CP/M
24.5 lb (11 kg)
$ 1 795



Dispositivos móviles

Compaq Portable

Compaq Computer Corporation, November 1982

Primer clon del IBM PC de 1981

8088 4.77MHz

128KB

9″ monocromo

Discos 5¼″

MS-DOS

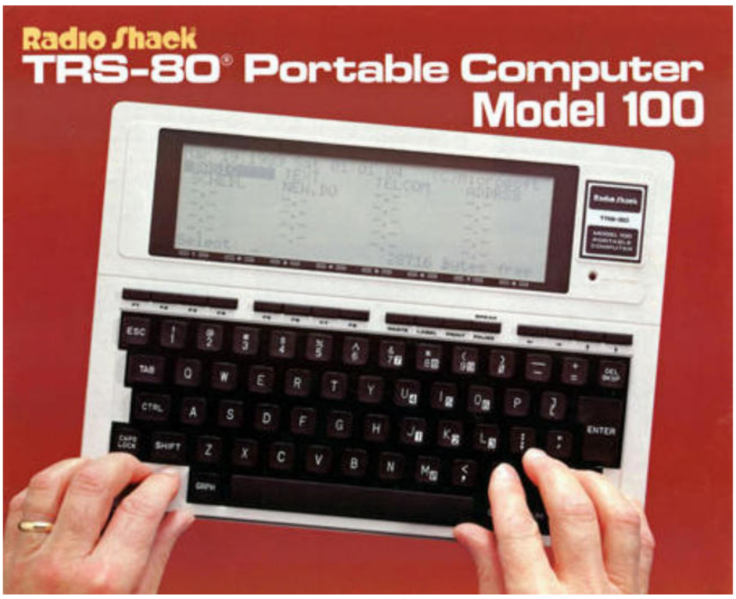
28 lb (12.7 kg)

$ 3 590

Dispositivos móviles

TRS-80 Model 100

Kyocera [Kyoto Ceramics], 1983



80C85 2.4MHz

16KB/32KB

LCD

BASIC in ROM

4 pilas AA

3.8 lb (1.7 kg)

$ 799

Dispositivos móviles

Apple Newton

Apple H1000 MessagePad, 1993

PDA [Personal Digital Assistant]

ARM 610 (RISC)

20MHz

640KB RAM

4MB ROM

LCD 336x240

0.41 kg

$ 699.99

Dispositivos móviles

PalmPilot 1000

US Robotics, March 1996

Motorola 68328

16MHz

128KB

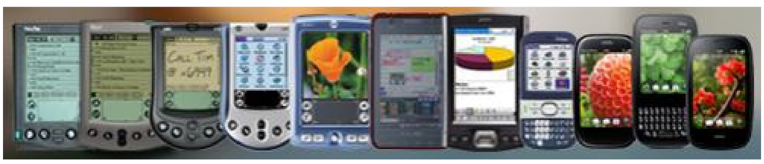
Palm OS

LCD 160x160

120x80x18mm

160 g (5.6 oz)

$ 299



Dispositivos móviles

IBM Simon

IBM Personal Communicator, 1994

Vadem x86

16MHz

1MB

LCD 160x293

200x64x30mm

510 g (18 oz)

$ 899

Dispositivos móviles

BlackBerry 950 Wireless Handheld

Research In Motion, January 1999

Intel 80386

512KB RAM

4MB flash

LCD 132x65

Modem 900MHz

1 batería AA

63x89x23mm

134 g

$ 399

Dispositivos móviles

iPhone

Apple, June 2007

Samsung ARM11 32-bit RISC

412MHz

128MB RAM

4GB flash

3.5" LCD 320x480 (163ppi)

iPhone OS 1.0

115x61x11.6mm

135 g

$499 - $599

Dispositivos móviles

iPad

Apple, April 2010

1GHz Apple A4 SoC

256MB RAM

16GB flash

9.7" LCD 1024x768 (163ppi)

iOS 3.2 → iOS 5.1.1

243x190x13mm

680 g

$499 - $699

Dispositivos móviles

iOS

iOS1 (2007) – iOS7 (2013)

Dispositivos móviles

iOS

https://www.computerworld.com/article/2975868/apple-ios/the-evolution-of-ios.html

… iOS12 (2018) … iOS13 (2019) … iOS14 (2020)

Dispositivos móviles

Android Inc.

fundada en octubre de 2003

adquirida por Google en julio de 2005 ($50M)

Dispositivos móviles

nexus

Gama original de smartphones de Google

Dispositivos móviles

Pixel

Gama actual de smartphones de Google
2016-

Pixel (2016)
Android v7.1

Pixel 2 (2017)
Android v8

Pixel 3 (2018)
Android v9

Pixel 4 (2019)
Android v10

Dispositivos móviles

Evolución

v1 (2009) …
v8 Oreo (2017) – v9 Pie (2018) – v10 (2019) – v11 (2020)

Dispositivos móviles

Evolución desde el punto de vista del programador

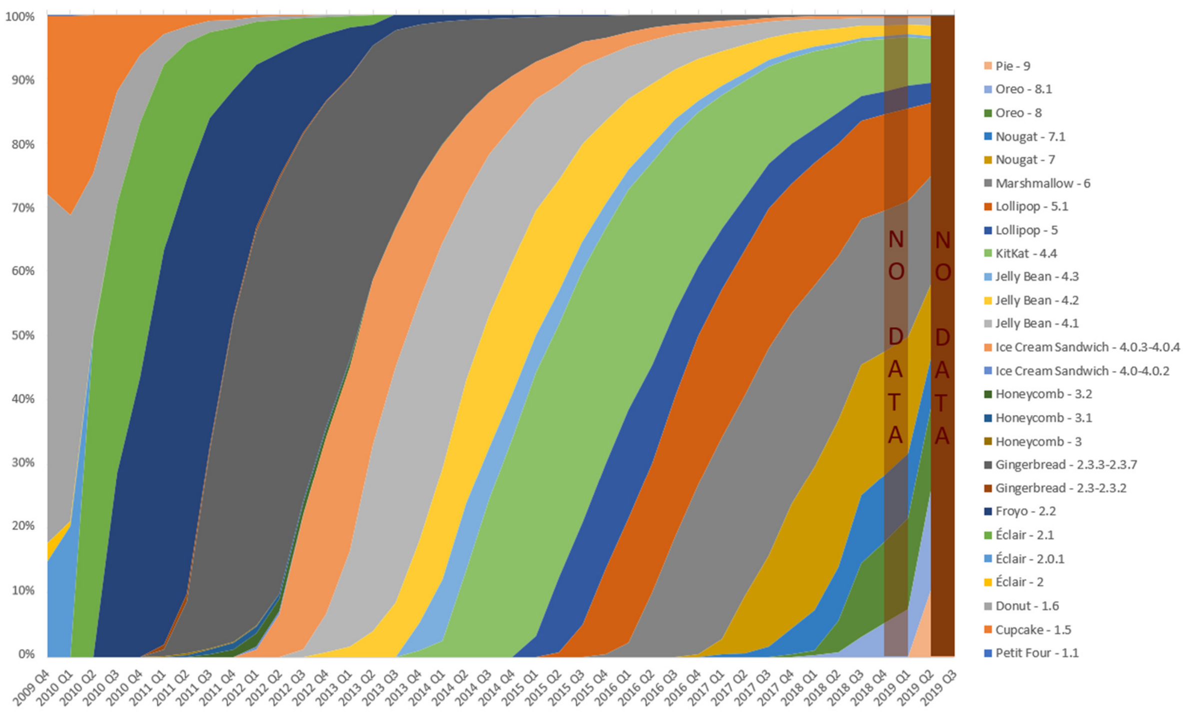
API Levels

Version	Marketing name	Release date	API level	Runtime
11	11	September 8, 2020	30	ART
10	10	September 3, 2019	29	ART
9	Pie	August 6, 2018	28	ART
8.1	Oreo	December 5, 2017	27	ART
8.0		August 21, 2017	26	ART
7.1	Nougat	October 4, 2016	25	ART
7.0		August 22, 2016	24	ART
6.0	Marshmallow	October 5, 2015	23	ART
5.1	Lollipop	March 9, 2015	22	ART
5.0		November 3, 2014	21	ART 2.1.0
4.4	KitKat	October 31, 2013	19	Dalvik (and ART 1.6.0)
4.3	Jelly Bean	July 24, 2013	18	Dalvik
4.2		November 13, 2012	17	Dalvik
4.1		July 9, 2012	16	Dalvik
4.0	Ice Cream Sandwich	October 19, 2011	15	Dalvik
2.3	Gingerbread	February 9, 2011	10	Dalvik 1.4.0

Dispositivos móviles

Distribución de versiones de Android

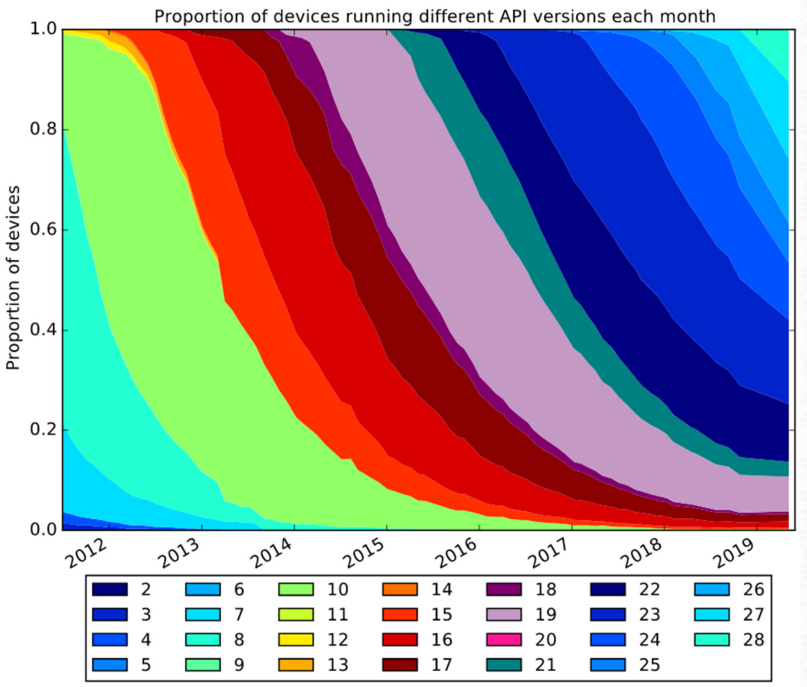
Datos de Google Play Store



Dispositivos móviles

Distribución de versiones de Android

Datos de androidvulnerabilities.org



Dispositivos móviles

Distribución de versiones de Android

Datos de Android Studio (2020)

Android Platform/API Version Distribution

ANDROID PLATFORM VERSION		API LEVEL	CUMULATIVE DISTRIBUTION
4.0	Ice Cream Sandwich	15	
4.1	Jelly Bean	16	99.8%
4.2	Jelly Bean	17	99.2%
4.3	Jelly Bean	18	98.4%
4.4	KitKat	19	98.1%
5.0	Lollipop	21	94.1%
5.1	Lollipop	22	92.3%
6.0	Marshmallow	23	84.9%
7.0	Nougat	24	73.7%
7.1	Nougat	25	66.2%
8.0	Oreo	26	60.8%
8.1	Oreo	27	53.5%
9.0	Pie	28	39.5%
10.	Android 10	29	8.2%

The minimum SDK version determines the lowest level of Android that your app will run on.

You typically want to target as many users as possible, so you would ideally want to support everyone -- with a minimum SDK version of 1. However, that has some disadvantages, such as lack of features, and very few people use devices that old anymore.

Your choice of minimum SDK level should be a tradeoff between the distribution of users you wish to target and the features that your application will need.

Click each Android Version/API level for more information.

OK Cancel

Dispositivos móviles

- Múltiples sistemas operativos
  https://en.wikipedia.org/wiki/Mobile_operating_system
- Millones de apps: redes sociales, viajes, finanzas…
- **Acceso a los datos más sensibles de los usuarios**

Seguridad y privacidad

“Don’t panic”

Berkman Center's Berklett Cybersecurity Project

Harvard Law School, February 2016

“While increasingly pervasive cryptography in consumer devices may close some surveillance channels, plenty of other channels are opening up that allow law enforcement to continue to keep an eye on suspected criminals. Most of these new inroads, the report says, come courtesy of two other tech innovations that are dramatically changing the way we use consumer electronics: the cloud and the Internet of Things (IoT).”

https://cyber.law.harvard.edu/pubrelease/dont-panic/

Seguridad y privacidad

“Don’t panic”

Berkman Center's Berklett Cybersecurity Project

Harvard Law School, February 2016

Bruce Schneier: “We’re not being asked to choose between security and privacy. We’re being asked to choose between less security and more security,”…

“Ubiquitous encryption protects us much more from bulk surveillance than from targeted surveillance,” …

Seguridad y privacidad

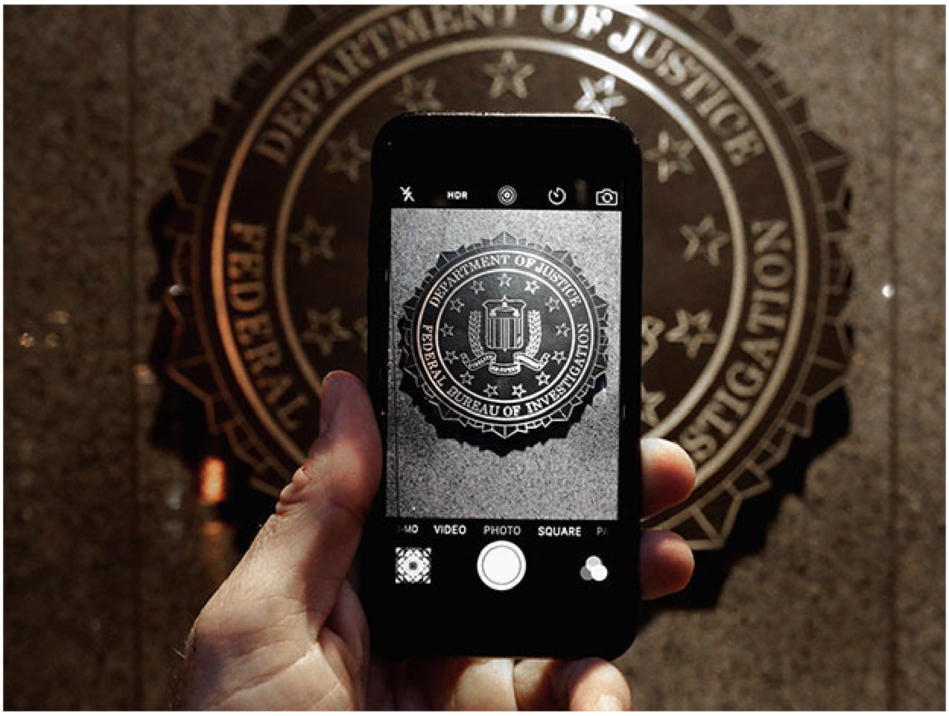
“Don’t panic”

Berkman Center's Berklett Cybersecurity Project

Harvard Law School, February 2016

“For a variety of technical reasons, computer security is extraordinarily weak. If a sufficiently skilled, funded, and motivated attacker wants in to your computer, they’re in. If they’re not, it’s because you’re not high enough on their priority list to bother with. Widespread encryption forces the listener—whether a foreign government, criminal, or terrorist—to [select a] target. And this hurts repressive governments much more than it hurts terrorists and criminals.”

Seguridad y privacidad



Seguridad y privacidad

1. The easy way in: Exploit a vulnerability in iOS 9

"Zero-day exploit" to switch off functions that thwarted the entry, e.g. a built-in delay that prohibits a user from trying too many incorrect password combinations at once, and an optional setting that prompts an iPhone to erase its memory after 10 failed entries.

Deployment: code sent as a malicious text message or by exploiting the driver that connects a charger to a laptop to enable new software to be uploaded to a phone

2. Trick the OS

Circumvent the iPhone's passcode protection by hijacking operations between the A6 and the non-volatile memory.

e.g. tamper with the physical line of communication that carries password recovery instructions between the two (to instruct the software to continue accepting failed passcode attempts until the investigators arrived at the correct one) & "brute force" attack.

3. Reset (and reset and reset) the memory

NAND Mirroring, i.e. remove the memory chip that NAND protects and make a digital copy of it. Once the copy is made, a hacker could test out combinations and simply reload the memory back onto the original chip before the 10-attempt limit is reached

e.g.
Hardware hack defeats iPhone passcode security

BBC, 19 September 2016

http://www.bbc.com/news/technology-37407047

https://youtu.be/tM66GWrwbsY

Seguridad y privacidad

4. Tear the whole thing apart

Physical attack in order to bypass certain tamper-resistant features, e.g. heating up the device in order to detach a memory chip, using acid to remove the surface layers of the chip in an act known as "decapping" & precision work with a tiny laser drill for reaching sections of the chip the hacker wants to more closely examine.

Goal: extract the handset's unique ID, which is a special digital key that Apple assigns to each device during manufacturing and could be used to decode an iPhone's memory.

Seguridad y privacidad

5. Sneak in through the side

Side channels: power consumption, acoustic properties, electromagnetic radiation, time it takes for a specific component to complete a task…

e.g. a hacker could hook up a resistor to the iPhone's internal circuits and read the amount of energy that flows by with each passcode attempt (Ben-Gurion University's Mimran likens it to putting your ear up to a safe, listening for a satisfying click as you turn the dial).

Seguridad y privacidad

Ejemplo: Axolotl

Keylogger for iPhone and Android

https://medium.com/@tomasreimers/axolotl-a-keylogger-for-iphone-and-android-a8b7b62cdab4

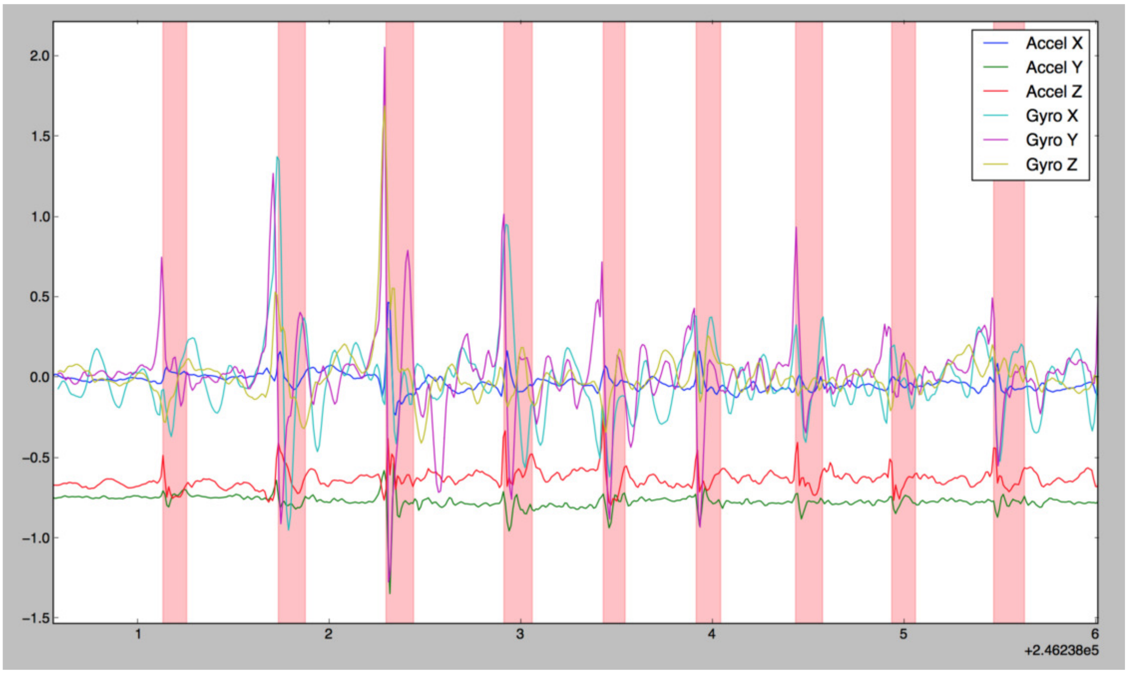
TL/DR: Descripción del ataque

- Lectura de datos de los sensores del dispositivo móvil (acelerómetro y giroscopio).
- Uso de técnicas de IA (aprendizaje automático) para predecir dónde ha pulsado el usuario la pantalla.
- Apps en segundo plano pueden acceder a los datos de los sensores mientras el usuario realiza otras tareas… tanto en iOS como en Android.

Seguridad y privacidad

Ejemplo: Axolotl

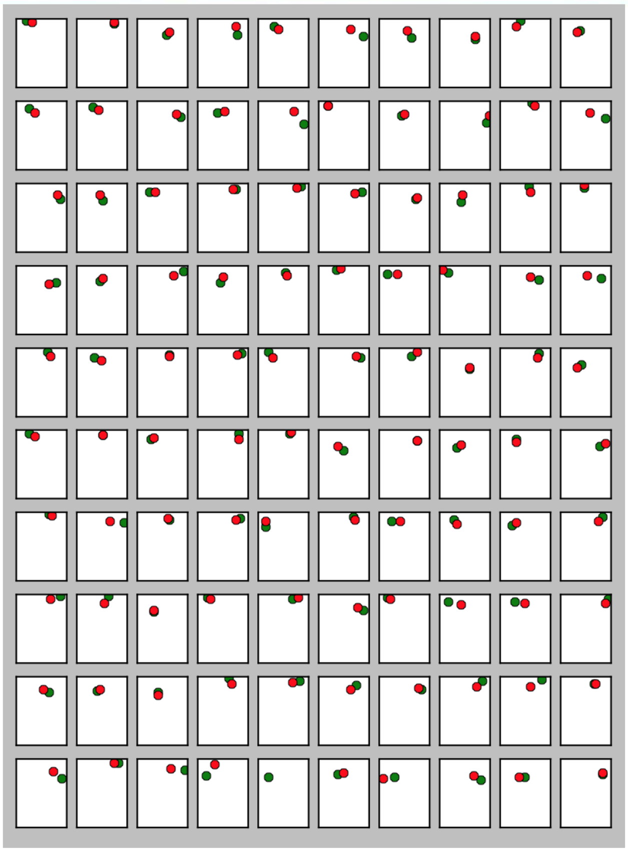
Lecturas de los sensores cuando se toca la pantalla



Seguridad y privacidad

Ejemplo: Axolotl

Predicción (en rojo) de dónde pulsa el usuario la pantalla (en verde)



https://github.com/tomasreimers/axolotl

Seguridad

Casos de uso

- **Electronic Flight Bag/Maintenance Tablet** – A device not typically connected to a network during normal operation.
- **Tethered Device** – A mobile device that relies on an external device for data communication and protection functions.
- **Secure Smartphone or Tablet** – Integrated devices, including their commercial wireless interfaces, applied to the warfighter's mission.
- **Multi-domain Device** – Single- or multi-user devices accessing multiple security domains.

Seguridad

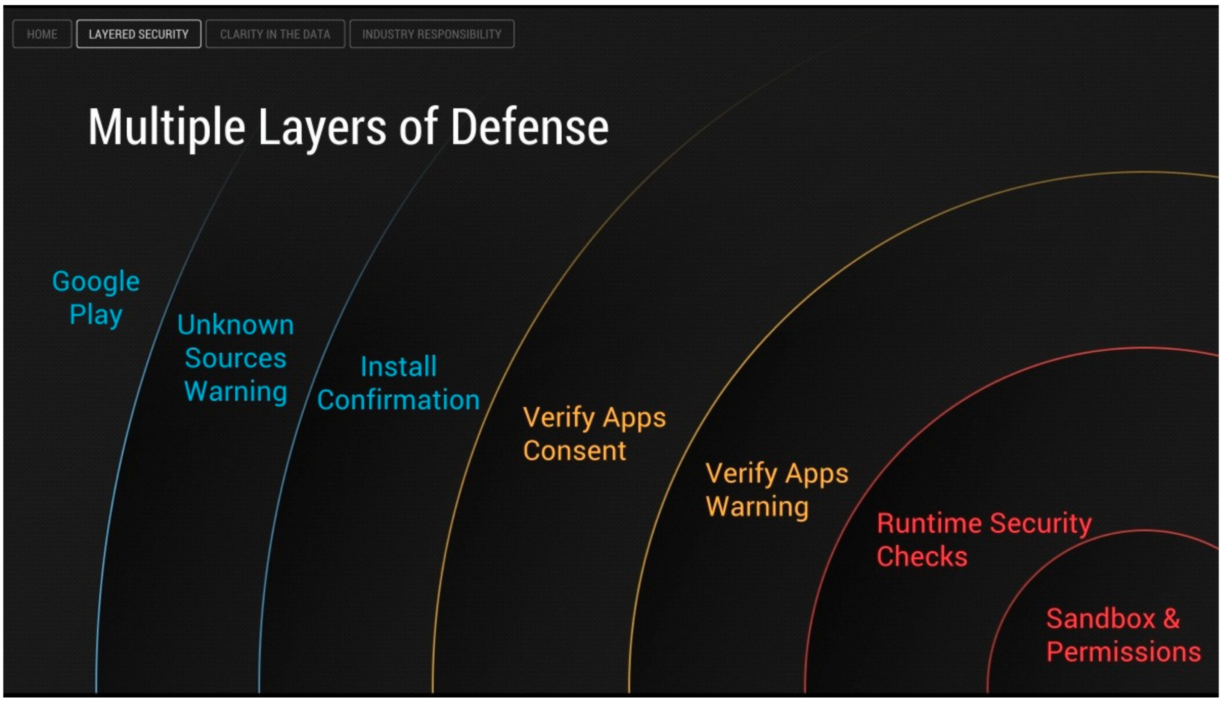
Casos de uso

Use Case	Security Protections				
	Device Resident Information	Received/ Transmitted Information	Identification and Auth. to Network	Network Protection	Multiple Domain Protection
EFB/Maint. Tablet	✓	✓			
Tethered Device	✓	partial	✓	✓	
Secure Smartphone/ Tablet	✓	✓	✓	✓	
Multi-domain Device	✓	✓	✓	✓	✓

http://www.embedded.com/design/safety-and-security/4398993/Securing-Android-for-warfare

Seguridad en Android

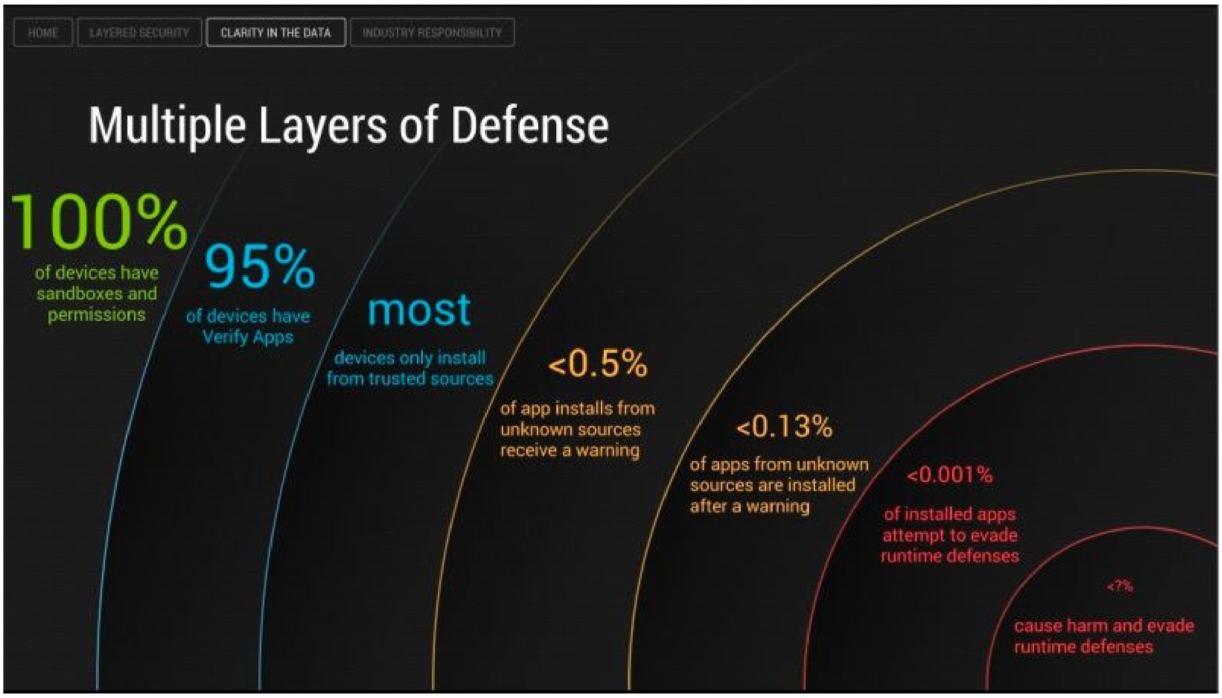
Múltiples capas de defensa en Android



http://lifehacker.com/how-secure-is-android-really-1446328680

Seguridad en Android

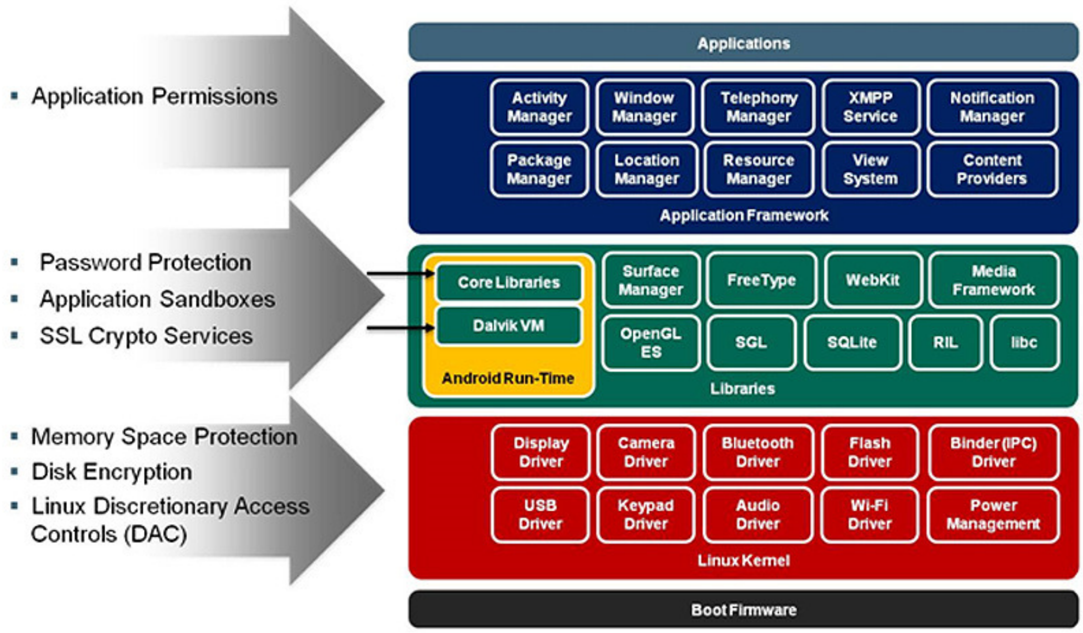
Múltiples capas de defensa en Android



http://lifehacker.com/how-secure-is-android-really-1446328680

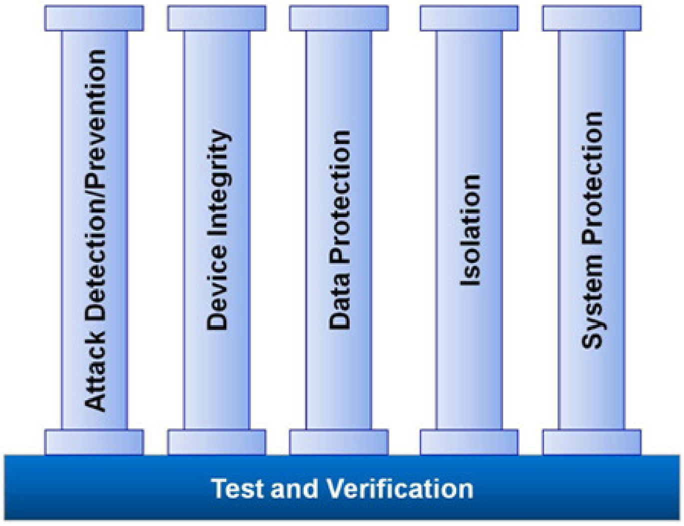
Seguridad en Android

El mercado de la seguridad en Android



http://www.embedded.com/design/safety-and-security/4398993/Securing-Android-for-warfare

Seguridad en Android



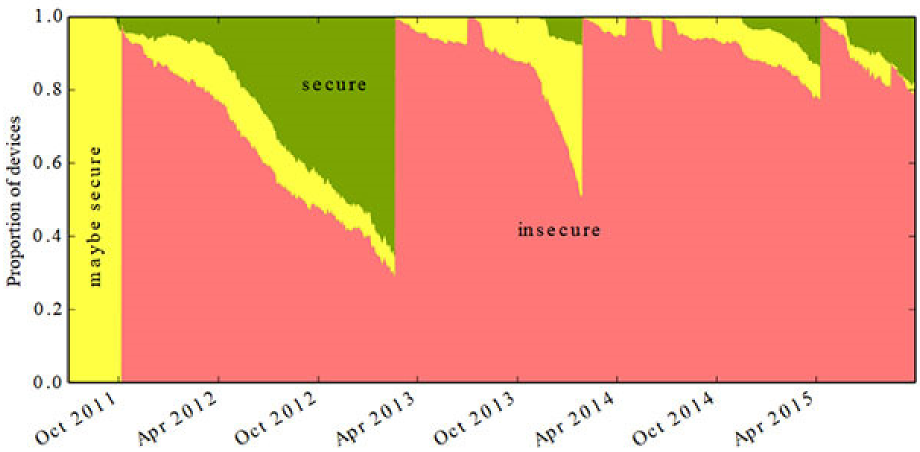
http://www.embedded.com/design/safety-and-security/4398993/Securing-Android-for-warfare

Seguridad en Android

Versiones inseguras de Android

http://androidvulnerabilities.org/

Proportion of devices running vulnerable versions of Android



This figure shows our estimate of the proportion of Android devices running *insecure*, *maybe secure* and *secure* versions of Android over time. Further details on how this figure constructed can be found on a separate page.

Seguridad en Android

Algunas vulnerabilidades conocidas de Android

Nickname	CVE or ID	Release (platform)	Cause of Vulnerability	Vulnerable Component: Linux	Vulnerable Component: Driver	Vulnerable Component: Daemon
asroot	2009-2692	08/2009 (≤2.2)	Null pointer dereference	socket	-	-
exploid	2009-1185	07/2010 (≤2.1)	Incorrect input validation	-	-	udev
RAtC	2010-EASY	10/2010 (≤2.2)	Incorrect error handling	-	-	adbd
Zimperlich	2010-EASY	12/2010 (≤2.2)	Incorrect error handling	-	-	zygote
KITNO	2011-1149	01/2011 (≤2.2)	Incorrect sharing of resources	-	-	init
psneuter	2011-1149	01/2011 (≤2.2)	Incorrect sharing of resources	-	-	init
GingerBreak	2011-1823	04/2011 (2.1-2.3.3)	Incorrect input validation	-	-	vold
Zergrush	2011-3874	10/2011 (2.2-2.3.6)	Buffer overflow	-	-	vold
levitator	2011-1350,1352	11/2011 (2.3-2.3.5)	Improper bound check	-	PowerVR	-
mempodroid	2012-0056	01/2012 (4.0-4.0.4)	Improper permission check	mem_write	-	-
bin4ry	OSVDB 94059	09/2012 (4.0-4.0.4)	Symlink attack	-	-	adbd
diaggetroot	2012-4220,4221	11/2012 (2.3-4.2)	Integer overflow	-	diagchar	-
-	2013-2094	06/2013 (2.2-4.3)	Integer overflow	perf	-	-
FramaRoot	2013-6282	04/2014 (2.x-4.x)	Missing checks	get/put_user	-	-
TowelRoot	2014-3153	06/2014 (4.0-4.4)	Use-after-free	futex	-	-
GiefRoot	2014-4321,4322	12/2014 (4.0-4.4)	Missing checks	-	camera	-
PingPongRoot	2015-3636	08/2015 (≥4.3)	Use-after-free	net	-	-

Seguridad

Android & Linux 3.6+ (USENIX Security 2016)

TCP side channel vulnerability CVE-2016-5696

"… a weakness in the Transmission Control Protocol (TCP) of all Linux operating systems since late 2012 that enables attackers to hijack users' internet communications completely remotely."

https://www.youtube.com/watch?v=S4Ns5wla9DY

Tiempo necesario: 40-60 segundos

Tasa de éxito: 88%-97%

Seguridad

Android: Stagefright

All a hacker would have to do to gain access to your device is send you a text message containing a malware-infected media attachment. The worst part? You don't even have to open the text or view the media in some cases. Drake noted that the remote MMS attack makes use of six critical vulnerabilities in Android operating systems 2.2 or later. Depending on the chat client you use, you may not even see the text message before it infects you—for instance, if you use Hangouts, the app will decipher the code before you ever get a notification of the text.

https://www.wired.com/2015/07/hack-brief-android-text-attack/

https://en.wikipedia.org/wiki/Stagefright_(bug)

Seguridad

iOS (iPhone & iPad)

"SandScout:
Automatic Detection of Flaws in iOS Sandbox Profiles"

ACM Conference on Computer and Communications Security, October 2016

"… the first systematic analysis of the iOS container sandbox profile. We propose the SandScout framework to extract, decompile, formally model, and analyze iOS sandbox profiles as logic-based programs. We use our Prolog-based queries to evaluate file-based security properties of the container sandbox profile for iOS 9.0.2 and discover seven classes of exploitable vulnerabilities. These attacks affect nonjailbroken devices running later versions of iOS. We are working with Apple to resolve these attacks…"

Seguridad

Configuración de Android

La seguridad de los dispositivos móviles depende no sólo del hardware y del software, sino también de la forma en que los usuarios interactúan y configuran el dispositivo.

De hecho, la configuración incorrecta del dispositivo es una de las principales causas de problemas de seguridad en dispositivos móviles.

T. Xu et al., "Do Not Blame Users for Misconfigurations," *Proc. 24th ACM Symp. Operating Systems Principles* (SOSP 13), 2013, pp. 244–259.

Seguridad

Configuración de Android

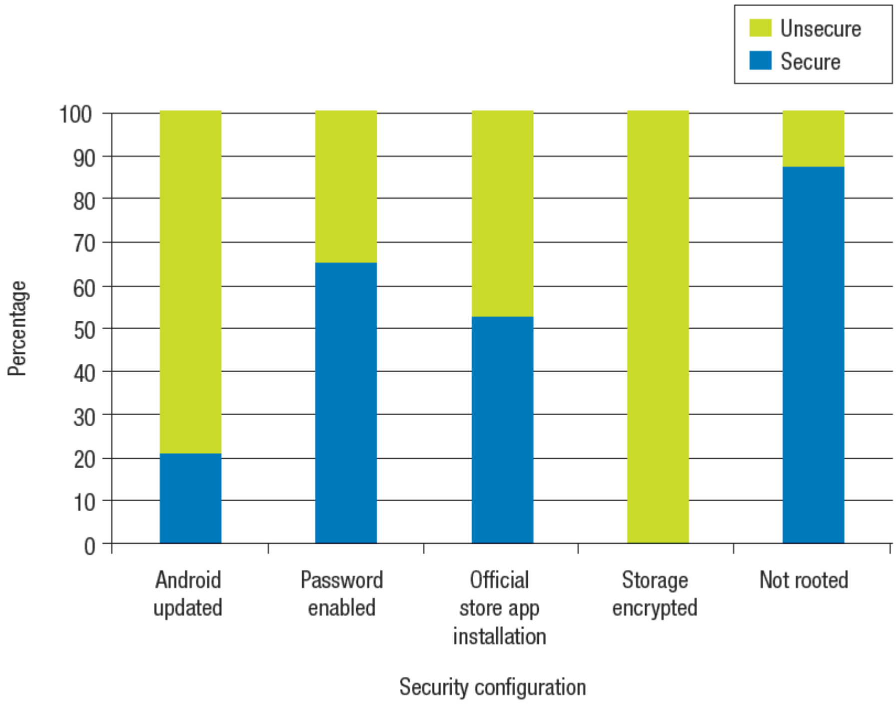
"The Perils of Android Security Configuration"
IEEE Computer, June 2016, pp. 15-21

Recomendaciones básicas:

1. Contraseña para desbloquear el teléfono.
2. Encriptar almacenamiento de datos (PIN).
3. Permitir sólo apps de Google Play Store.
4. Mantener actualizado el sistema operativo ☹
5. No "rootear" el dispositivo [a.k.a. jailbreaking].

Seguridad

Configuración de Android



Seguridad

Configuración de Android

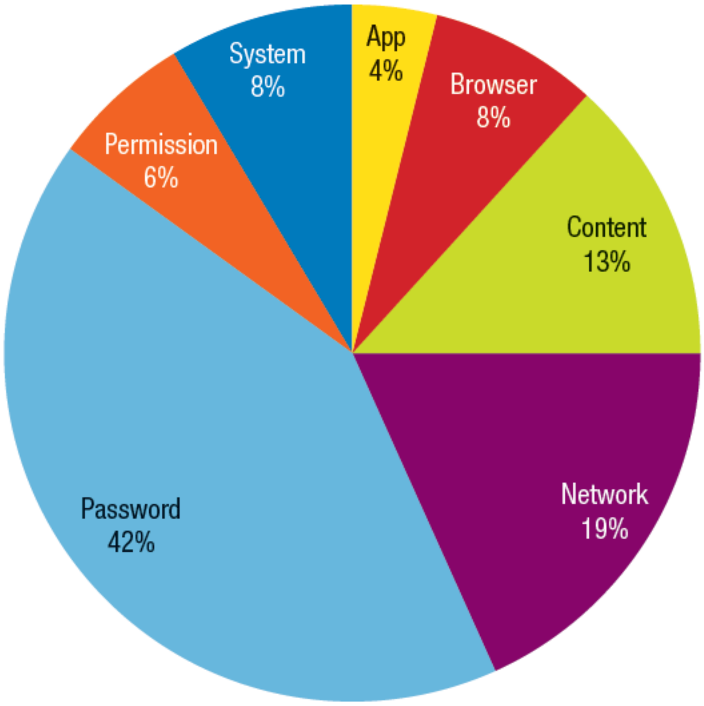
CIS [Center for Internet Security]

41 opciones de configuración agrupadas en 7 categorías:

1. Aplicaciones, p.ej. actualizaciones automáticas
2. Navegador, p.ej. uso de JavaScript y cookies
3. Contenido, p.ej. datos almacenados en el dispositivo
4. Red, p.ej. Bluetooth & Wi-Fi
5. Uso de contraseñas "seguras"
6. Permisos, p.ej. acceso a los ficheros del sistema
7. Sistema, p.ej. uso de GPS & instalación de ficheros .apk de fuentes desconocidas

Seguridad

Configuración de Android



Seguridad

Configuración de Android

TABLE 1. Misconfigurations per Center for Internet Security category: Android devices with the default factory configuration versus the best and worst configurations.

Configuration	No. of misconfigurations per possible settings (% of misconfigurations)							
	Application	Browser	Content	Network	Password	Permission	System	Total
Worst	3/3 (100)	2/3 (67)	4/6 (67)	5/8 (62)	9/12 (75)	3/4 (75)	3/5 (60)	**29/41 (71)**
Factory	1/3 (33)	2/3 (67)	0/6 (0)	5/8 (62)	9/12 (75)	1/4 (25)	1/5 (20)	**19/41 (46)**
Best	1/3 (33)	2/3 (67)	3/6 (50)	2/8 (25)	6/12 (50)	1/4 (25)	0/5 (0)	**15/41 (37)**

"Android manufacturers should provide better security settings in advance to protect more users by default."

Seguridad

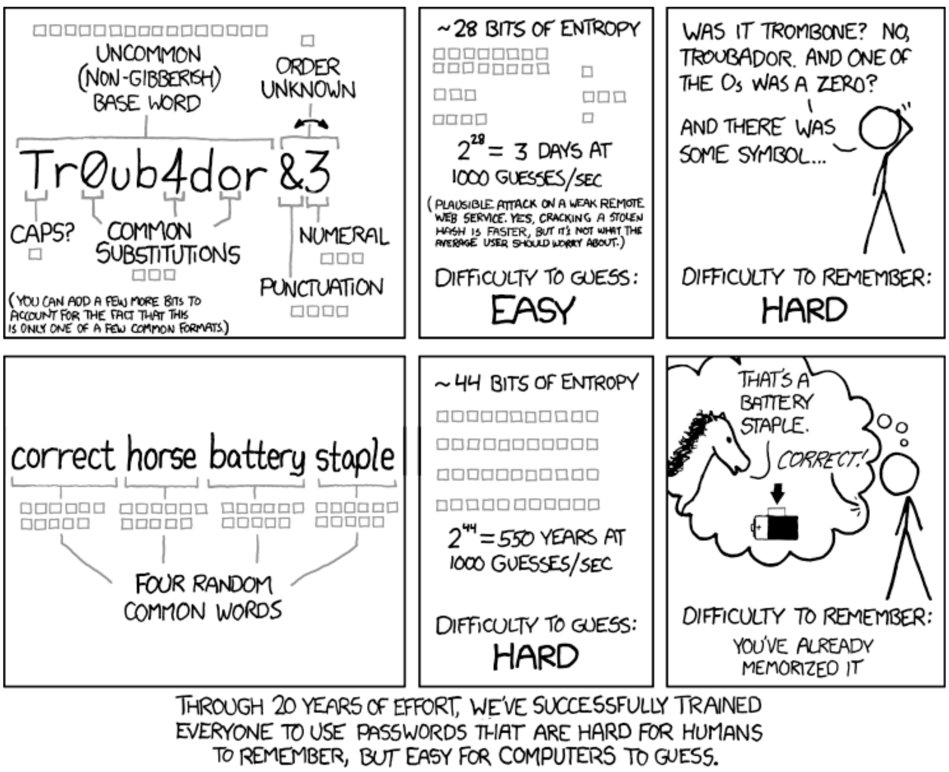
Configuración de Android:
¿Contraseñas seguras?

p@ssw0rd	pAsswOrd
punk4life	punkforlife
ieatkale88	iloveyou88
jonnyrtxe	jonny1421
sk8erboy	skaterboy
1qaz2wsx3edc	thefirstkiss

Do Users' Perceptions of Password Security Match Reality?, CHI'2016

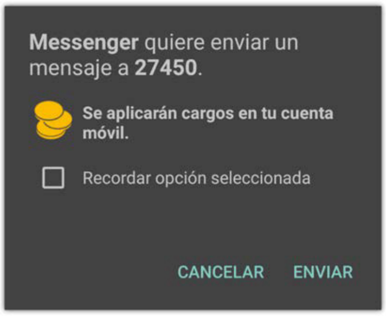
Autentificación

¿Contraseñas seguras? https://xkcd.com/936/



Seguridad

Configuración de Android:
Instalación de fuentes desconocidas
p.ej. “WhatsApp Trendy Blue” en mercados no oficiales



PELIGRO: Hay que acordarse de volver a activar la prohibición de instalación de fuentes no oficiales después de haberlo permitido puntualmente por algún motivo justificado (p.ej. Pokémon Go)

Seguridad

Configuración de Android:
Android Jailbreaking

“Android root is the voluntary and legitimate process of gaining the highest privilege and full control over a user's Android device.”

- **Cons:** unwanted access, data leaks, theft & voiding of the manufacturer warranty.
- **Pros:** install the most recent Android version, delete unwanted built-in applications & customize UI.

Seguridad

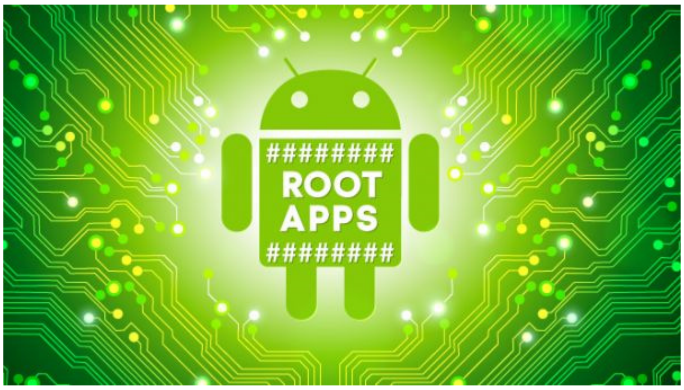
Configuración de Android:
Android Jailbreaking

Android soft root methods:

- Most public Android root exploits target the application-layer vulnerabilities that affect only specific types of devices.
- Although kernel vulnerabilities are considered the most dangerous, an exploit developed on one device may need to be adapted to work on another.
- As kernel vulnerabilities become rare, device drivers become the dominating target to find root exploits.

Seguridad

Android rooting (a.k.a. jailbreaking)
more than 160 exploits, subcategorized into 59 families



Hang Zhang, Dongdong She & Zhiyun Qian:
"Android Root and its Providers: A Double-Edged Sword"
22nd ACM SIGSAC Conference on Computer and Communications Security (CCS '15), Denver, Colorado, October 2015, 1093-1104.
http://dx.doi.org/10.1145/2810103.2813714

Android rooting (a.k.a. jailbreaking)

e.g.

TacoRoot (CVE-2014-3153) exploits a World-writable recovery log file /data/data/recovery/log. Attacker can symlink it to /data/local.prop, reboot to recovery-mode and write a new log to set ro.kernel.qemu=1, which yields the root privilege

Impacto: Dispositivos Android de 2011-2012

Android rooting (a.k.a. jailbreaking)

Exploits: Linux Kernel

e.g.

TowelRoot (CVE-2014-3153): futex syscall
(a method for waiting until a certain condition becomes true, typically used as a blocking construct in the context of shared-memory synchronization).

Impacto: Todos los dispositivos con kernel Linux <3.14.5

https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2014-3153

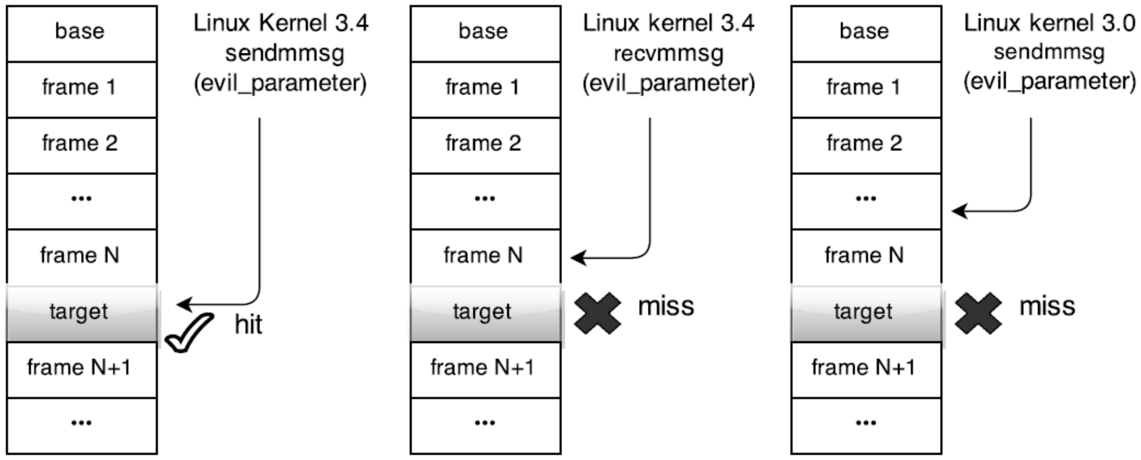
3 de junio de 2014

Seguridad

Android rooting (a.k.a. jailbreaking)

Exploits: Linux Kernel

Futex bug (CVE-2014-3153)



kernel stack overwrite by invoking system calls

Seguridad

Android rooting (a.k.a. jailbreaking)

Exploits: Vendor-Specific Kernel or Drivers

- Qualcomm's custom Linux kernel branch
- Vendor-specific device drivers for various peripherals (e.g., camera, sound), e.g. some Samsung devices.

Impacto: Un subconjunto de dispositivos Android (p.ej. Una gama determinada de móviles, tablets o e-books).

Seguridad

Android rooting (a.k.a. jailbreaking)

Exploits: Libraries Layer.

- ZergRush exploit (CVE-2011-3874): libsysutils used by Volume Manager daemon (running as root) in Android is shown to have a stack overflow vulnerability that leads to root privilege escalation.
- ObjectInputStream vulnerability (CVE-2014-7911).

Impacto: Elevado, ya que la biblioteca puede utilizarse en muchas aplicaciones que se ejecuten como root.

Seguridad

Android rooting (a.k.a. jailbreaking)

Exploits: Application and Application Framework

e.g. vulnerable logics introduced by setuid utilities, system applications, or services.

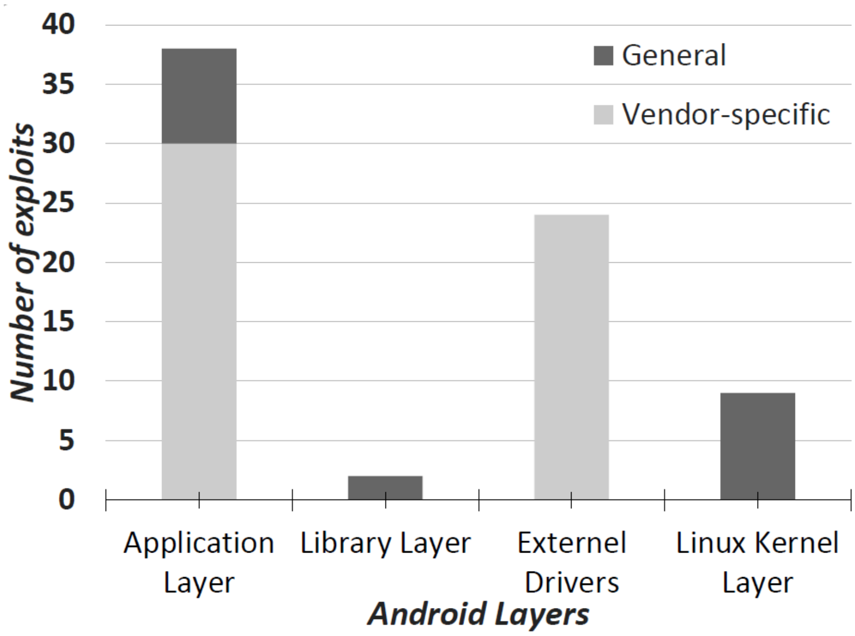
- Backdoor-like setuid binary shipped with certain ZTE Android devices (CVE-2012-2949).

Impacto: Más limitado, siempre y cuando afecte a aplicaciones no demasiado comunes.

Seguridad

Android rooting (a.k.a. jailbreaking)

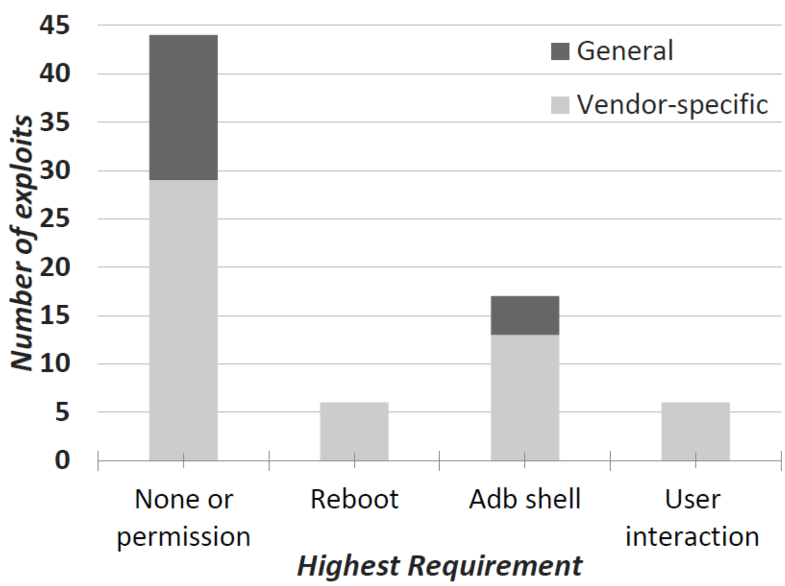
Exploits por categoría



Seguridad

Android rooting (a.k.a. jailbreaking)

Exploits: ¿Requieren la colaboración del usuario?



Seguridad

Android rooting (a.k.a. jailbreaking)

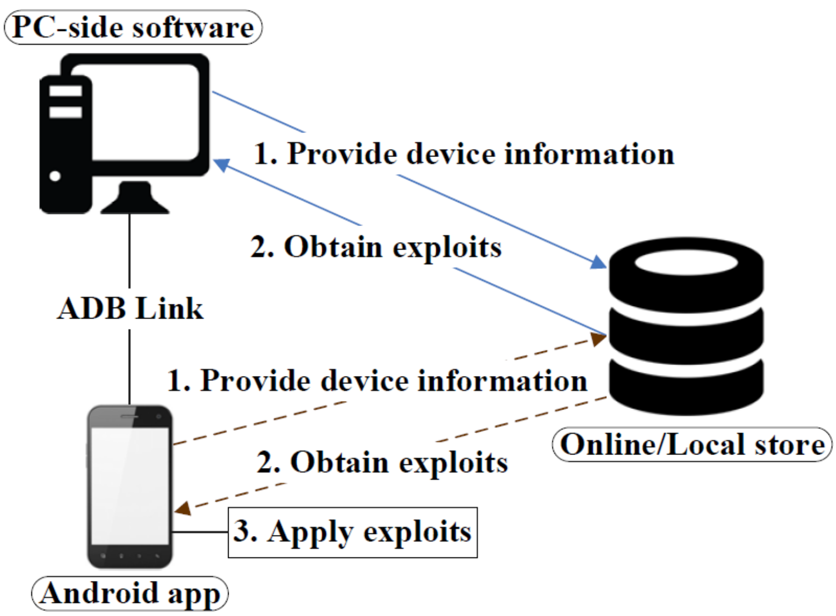
Exploits: ¿Funcionan los antivirus?

Root exploit	AVG	Lookout	Norton	Trend Micro
exploid(2010)				
Zimperlich(2010)	X	X		
Gingerbreak(2011)	X	X	X	X
BurritoRoot(2012)	X	X		X
Poot(2013)				X
LGPwn(2013)			X	X
WeakSauce(2014)	X	X		
Framaroot(2014)	X			X
Towelroot(2014)	X	X	X	X
PingPong root(2015)				X

Seguridad

Android rooting (a.k.a. jailbreaking)

Exploits: Root providers



Seguridad

Android rooting (a.k.a. jailbreaking)

Exploits: Root providers

Name	Components	Devices supported (claimed)
Root Genius	PC/MOBI	20,000+
360 Root	PC/MOBI	20,000+
IRoot	PC/MOBI	10,000+
King Root	PC/MOBI	10,000+
SRSRoot	PC	7,000+
Baidu Root	PC/MOBI	6,000+
Root Master	PC/MOBI	5,000+
Towelroot	MOBI	N/A
Framaroot	MOBI	N/A

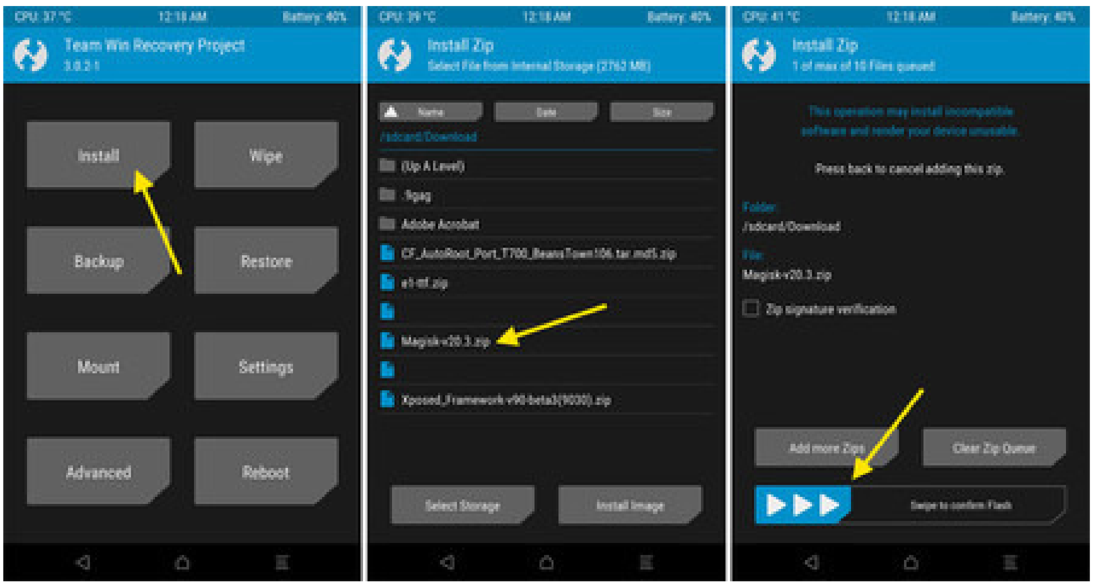
Muchos han dejado de actualizarse…

Seguridad

Android rooting (a.k.a. jailbreaking)

Exploits: Root providers

Magisk (2020)

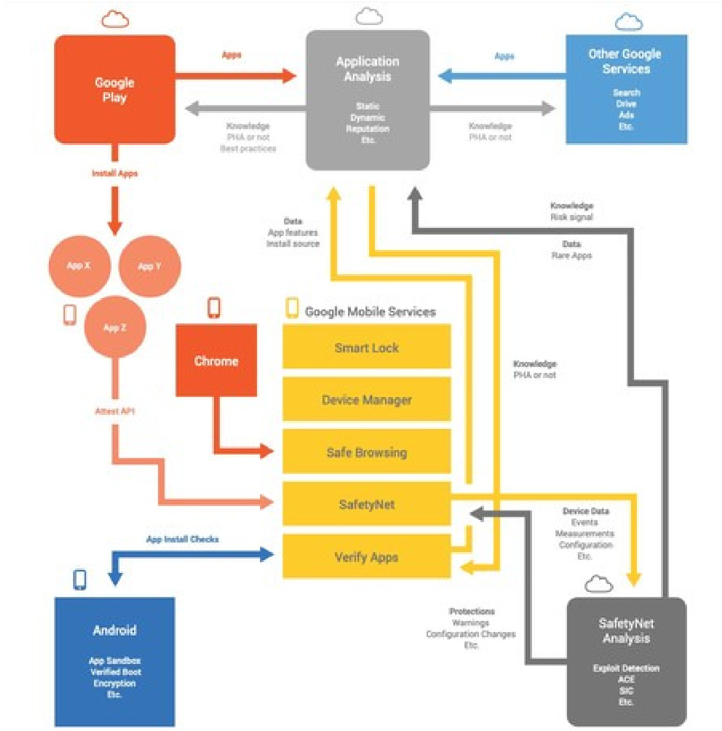


Seguridad

Android rooting (a.k.a. jailbreaking)

SafetyNet

Certificación por hardware (2020)



Impide el uso de herramientas como Magisk…

Seguridad

Configuración de Android: Herramientas de evaluación

p.ej.

uSEA

User-defined Security Configuration Assessment tool

- Identifica problemas de configuración en dispositivos Android usando las recomendaciones del CIS.
- Mantiene una base de datos en la nube para analizar los problemas de configuración más habituales.

D. Vecchiato, M. Vieira & E. Martins, "A Security Configuration Assessment for Android Devices," *Proc. 30th Ann. ACM Symp. Applied Computing* (SAC 15), 2015, pp. 2299–2304.

Seguridad

Herramientas MDM

Mobile Device Management

- AirWatch (VMware)
  http://www.airwatch.com
- Lookout
  http://www.lookout.com

De forma remota, borran el contenido de los dispositivos y los bloquean, además de monitorizar su localización.

Seguridad

Herramientas

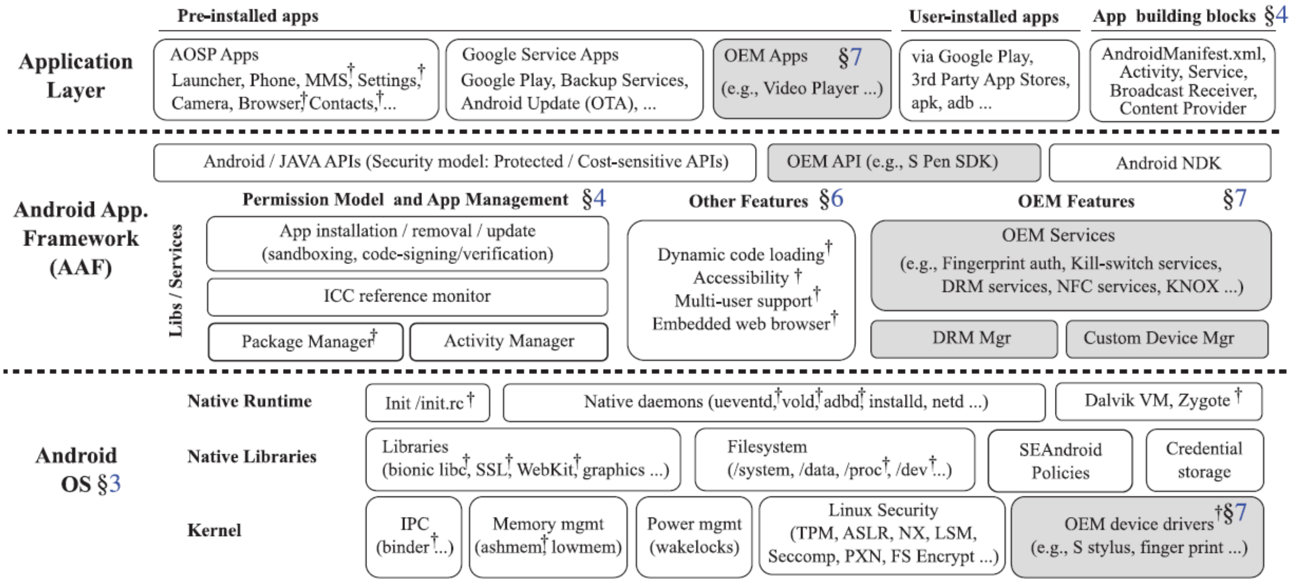
CCNDroid

Herramientas de seguridad desarrolladas por el CCN-CERT para dispositivos con sistema operativo Android.

- **CCNDroid Wiper**: Herramienta para el borrado seguro de ficheros.
- **CCNDroid Crypter:** Herramienta para el cifrado de ficheros con distintos algoritmos (incluido PGP).

https://www.ccn-cert.cni.es/herramientas-de-ciberseguridad/ccndroid-publico.html

Android software stack in terms of security



Android software stack in terms of security

Android OS (1/2): The Linux kernel is the foundation of the whole software stack. Android implements the application-level sandbox by leveraging Linux's Discretionary Access Control (DAC). By assigning a unique uid to each app, Android isolates individual apps within a uid-based process boundary. Therefore, an app cannot interact with other apps by default and can only access resources in its own sandbox (e.g., own files). Similarly, each system resource (e.g., network, sound, etc.) is assigned a unique gid: to grant an app access to a particular resource, the app's uid is added to the resource's gid group.

Seguridad

Android software stack in terms of security

Android OS (2/2): Although many Android apps are running in the Dalvik Virtual Machine (VM), the VM does not provide additional sandboxing like the Java VM does, so the only security boundary of an Android app is the DAC-based application sandbox.

Attacks at this layer mainly focus on breaking the DAC sandbox by exploiting particular **kernel vulnerabilities**, while defensive techniques focus on hardening the kernel to either eliminate the vulnerability or reduce impact when exploits occur.

Seguridad

Android software stack in terms of security

Android Application Framework (AAF):

Abstracted from the Linux DAC model, in order to provide apps fine-grained accesses to resources (such as GPS or contacts), Android implements its **permission model** at the AAF. To gain such permissions, the developer first declares the resources required for his app; then, the user approves the declared permissions upon app installation.

Seguridad

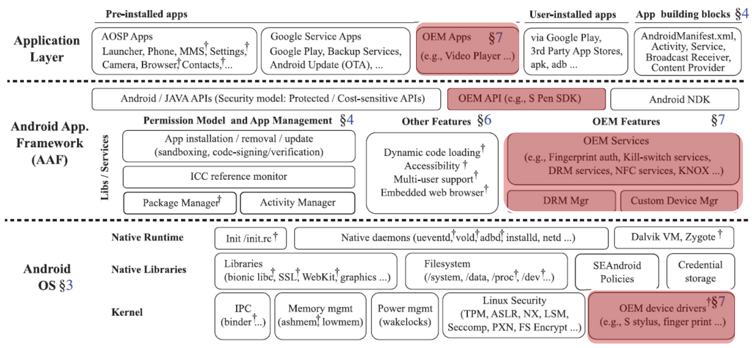
Android software stack in terms of security

Application Layer:

- **AndroidManifest.xml** file: App Activities, Services, BroadcastReceivers, and ContentProviders.
- App components communicate through Intent, the default Intercomponent Communication (ICC) channel, and share their data with other apps using Content Provider (e.g. SQLite).
- App developers also have the freedom to build apps in native code…

Seguridad

Android software stack in terms of security



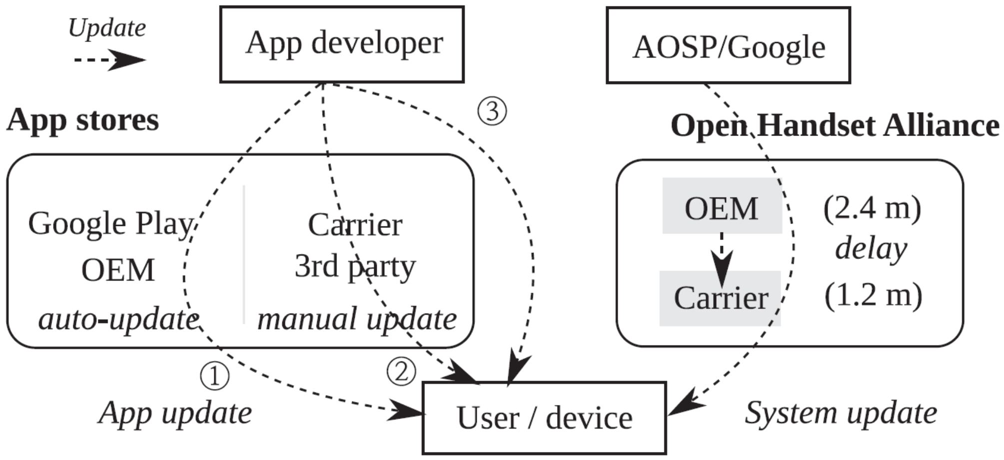
Device fragmentation:

- Highly customized componentes by device distributors like OEMs and carriers.
- However, such customization frequently introduces new security issues :-(

Sólo Google: 147 builds de Android (v1.6-v.6.0)
>24K dispositivos diferentes

Seguridad

Android software stack in terms of security

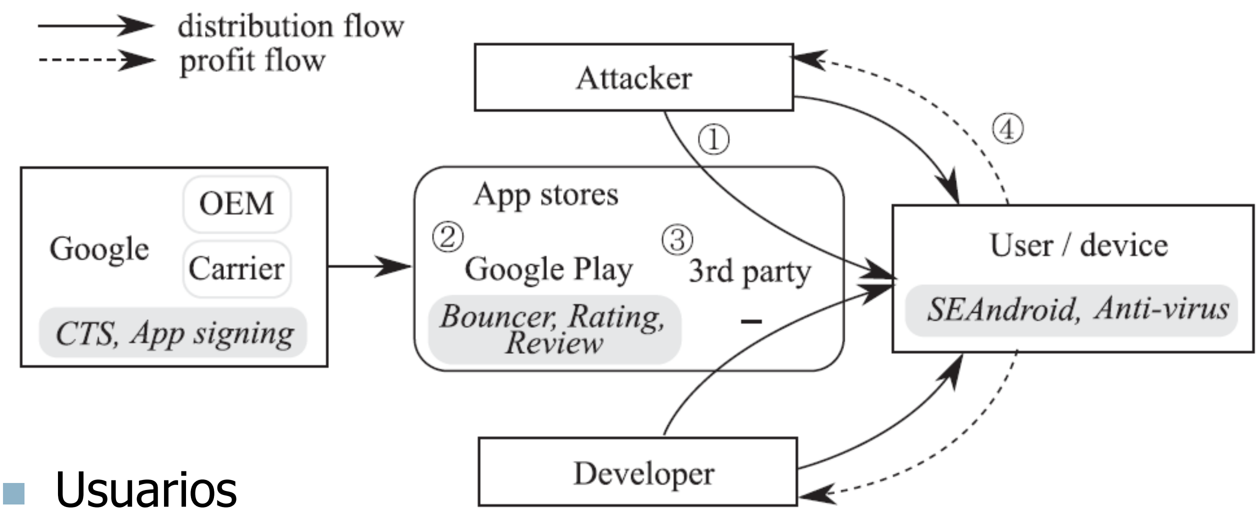


Device fragmentation:

Proceso de actualización (cuando existe)

Seguridad

"Stakeholders" en el ecosistema Android



- Usuarios
- Desarrolladores
- App stores
- Open Handset Alliance (Google, OEMs & carriers)

"Stakeholders" en el ecosistema Android

Desarrolladores de malware

Monetization Scheme	Description	Financial Benefit
Toll fraud	Premium rate number billing via SMS or call	Direct
Click fraud	Imitating user's clicks in pay-per-click ads	Direct
Pay-per-installation	Payment per app installation (e.g., up to $1 USD)	Direct
Ad profit hijacking	Replacing developer's ID to reroute ad income	Direct
Ad network hijacking	Replacing the underlying ad provider	Direct
Adware	Harvesting ad views	Direct
Mining	Mine virtual currency on user's devices	Direct
IMEI/IMSI stealing	Stealing device identity (e.g., imitate or unblock)	Indirect
Spyware	Stealing personal information (e.g., contacts)	Indirect
Man-in-the-mobile	Stealing random tokens (e.g., mTAN for Bank app)	Indirect
Ranking poisoning	Requesting fake search queries to boost rank	Indirect

Androidismos

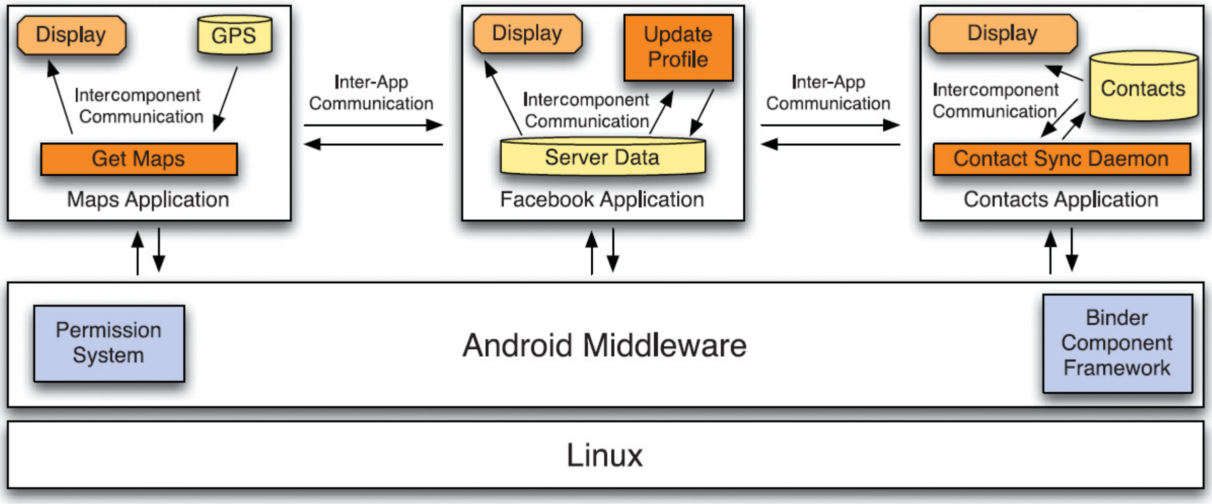
PECULIARIDADES DE ANDROID

- Aunque las aplicaciones se programan en Java, se empaquetan en **ficheros APK** que no contienen Java bytecodes: el SDK de Android transforma los bytecodes Java en **bytecodes DEX** [Dalvik executable].
- Existen herramientas de "retargeting" que transforman bytecodes DEX a bytecodes Java (para aprovechar las herramientas de análisis ya disponibles para Java), p.ej. ded, Dare, dex2jar…

Seguridad

Androidismos

PECULIARIDADES DE ANDROID



Seguridad

Androidismos

PECULIARIDADES DE ANDROID

- Ciclo de vida de las aplicaciones (modelo de ejecución)
- Componentes clave
  - Actividades
    (componentes UI)
  - Servicios
    (ejecución en segundo plano)
  - Content provider
    (acceso a almacenes de datos)
  - Broadcast receiver
    (recepción de notificaciones de eventos)

Seguridad

Androidismos

PECULIARIDADES DE ANDROID

Comunicación entre componentes (ICC):

- Intents (paso de mensajes):
  - Destinatario específico
  - "Action strings" & "intent filters"
  - "extras" (key-value Bundle)
- Comunicación entre procesos: Binder IPC.
- Interacción con el sistema operativo: Eventos.
  - System callbacks
  - UI callbacks

Seguridad

Androidismos

PECULIARIDADES DE ANDROID

Almacenamiento de datos:

- Suspensión de actividades (en cualquier momento)
- Ficheros
- SharedPreferences (key-value storage)
- Content providers, p.ej. SQLite

Herramientas de análisis

***droid: Assessment and Evaluation of Android Application Analysis Tools**

BRADLEY REAVES and JASMINE BOWERS, University of Florida
SIGMUND ALBERT GORSKI III, North Carolina State University
OLABODE ANISE, RAHUL BOBHATE, RAYMOND CHO, HIRANAVA DAS, SHARIQUE HUSSAIN, HAMZA KARACHIWALA, NOLEN SCAIFE, BYRON WRIGHT, and KEVIN BUTLER, University of Florida
WILLIAM ENCK, North Carolina State University
PATRICK TRAYNOR, University of Florida

The security research community has invested significant effort in improving the security of Android applications over the past half decade. This effort has addressed a wide range of problems and resulted in the creation of many tools for application analysis. In this article, we perform the first systematization of Android security research that analyzes applications, characterizing the work published in more than 17 top venues since 2010. We categorize each paper by the types of problems they solve, highlight areas that have received the most attention, and note whether tools were ever publicly released for each effort. Of the released tools, we then evaluate a representative sample to determine how well application developers can apply the results of our community's efforts to improve their products. We find not only that significant work remains to be done in terms of research coverage but also that the tools suffer from significant issues ranging from lack of maintenance to the inability to produce functional output for applications with known vulnerabilities. We close by offering suggestions on how the community can more successfully move forward.

CCS Concepts: • **Security and privacy → Software and application security;** • **Software and its engineering → Automated static analysis; Dynamic analysis;**

55

"*droid: Assessment and Evaluation of Android Application Analysis Tools", ACM Computing Surveys, October 2016

Herramientas de análisis

PREGUNTAS

- ¿Qué áreas de seguridad han sido investigadas y cuáles requieren todavía más atención?
- ¿Qué herramientas están disponibles para que los desarrolladores de aplicaciones produzcan apps más seguras?

Seguridad

Herramientas de análisis

DIFICULTADES DE JAVA

- Herencia y polimorfismo
- Reflexión
- Carga dinámica de código
- Ejecución de código nativo (JNI)

… y todo eso si sólo nos fijamos en el análisis estático de las aplicaciones (el análisis dinámico es aún más difícil).

Seguridad

Herramientas de análisis estático

		FlowDroid [Arzt et al. 2014]	DroidSafe [Gordon et al. 2015]	Amandroid [Wei et al. 2014]	MalloDroid [Fahl et al. 2012]	DroidSift [Zhang et al. 2014a]	Epicc [Octeau et al. 2013]	AdRisk [Grace et al. 2012]	UI Deception [Bianchi et al. 2015]	CryptoLint [Egele et al. 2013]	CLAPP [Fratantonio et al. 2015]	CHEX [Lu et al. 2012]	Blue Seal [Shen et al. 2014]	MassVet [Chen et al. 2015]	Stowaway [Felt et al. 2011]	PScout [Au et al. 2012]	Drebin [Arp et al. 2014]	DIDFail [Klieber et al. 2014]	Anadroid [Liang et al. 2013]	AppProfiler [Rosen et al. 2013]	AsDroid [Huang et al. 2014]	COPES [Bartel et al. 2012b]	SEFA [Wu et al. 2013]	AAPL [Lu et al. 2015]	Intent Input Validation Vulnerability Detection [Gallingani et al. 2015]	DroidEagle [Sun et al. 2015]	User-Aware Privacy Control Analysis [Xiao et al. 2012]	User-Centric Dependence Analysis [Elish et al. 2012]	ViewDroid [Zhang et al. 2014b]	DroidJust [Chen and Zhu 2015]	SCANDAL [Kim et al. 2012]	WeChecker [Cui et al. 2015]	AppIntent [Yang et al. 2013]	WiFiLeaks [Achara et al. 2014]	Comdroid [Chin et al. 2011]	Apposcopy [Feng et al. 2014]
Reflection	String constants	●	●	○	○	●	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○
	Constructed strings	○	●	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	Obfuscation/encryption handling	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Misc	Dynamic DEX file loading	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	Multiapp support	○	○	○	○	○	●	○	●	○	○	○	◐	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○
	Native code	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○
	Extensible	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○
	Publicly available	●	●	●	●	○	●	○	○	○		○	●	○	○	●	○	●	●	●	○	○	○	○	○		●	○	○	○	○	●	○	○	○	○
	Benign only/malicious/dual use	D	D	D	D	M	D	B	D	B	D	B	B	M	B	B	M	D	M	D	M	B	B	D	B	M	M	M	M	D	D	B	B	M	B	M
Vulnerabilities Detected	Detect sensitive information leaks	●	●	○	○	○	○	●	○	○	○	●	●	○	○	○	○	●	●	○	○	○	●	●	○	○	●	●	○	●	●	●	●	●	○	●
	Authentication	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	Cryptography	○	○	○	●	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	Data validation	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	Intent spoofing	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	●	○
	Unauthorized intent receipt	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	◐	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○
	Configuration and deployment	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	Permission misuse	○	○	○	○	○	○	○	○	○	○	○	●	○	●	●	◐	○	●	○	○	●	●	○	○	○	○	○	○	○	○	●	○	○	○	○
	Plagiarism detection	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	●	○	○	●	○	○	○	○	○	○	○
	Other	○	○	○	○	•	○	○	•	○	•	○	○	○	○	○	○	○	○	•	•	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Technique	Taint analysis	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○	●	●	○	●	○	○	●	○	●	●	○	○	●
	Reachability analysis	○	○	○	○	○	○	●	○	○	○	●	●	○	○	○	○	○	○	○	○	●	●	○	○	○	○	○	○	○	○	○	○	○	●	○
	Symbolic execution	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○	○	●	○	○	○
	APK rewriting	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	Intercomponent flow graph	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	Intercomponent flow analysis	○	○	○	○	○	●	●	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○
	Heuristics	○	○	○	●	○	○	○	●	●	○	○	○	○	●	●	○	○	○	○	○	○	●	○	○	○	○	●	○	○	○	○	○	●	○	●
	View comparison	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	●	○	○	●	○	○	○	○	○	○	○
	Other	○	○	○	○	•	○	○	•	•	•	○	○	○	○	○	•	○	•	•	•	•	○	•	○	○	○	○	○	○	•	○	○	○	○	○
Frameworks	Soot	●	●	○	○	●	●	○	○	○	○	○	●	○	○	●	○	●	○	○	○	●	○	○	●	○	○	●	○	●	○	●	●	○	○	●
	WALA	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	●	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○
	Androguard	○	○	○	●	○	○	○	●	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○
	Baksmali	○	○	○	○	○	○	●	○	○	○	○	○	●	○	○	○	○	●	○	○	○	●	○	○	○	○	○	●	○	○	○	○	○	○	○
	Other	○	○	•	○	○	○	○	○	○	○	•	○	○	•	○	○	•	○	•	○	○	○	•	○	○	•	○	○	○	○	○	○	○	•	○

Seguridad

Herramientas de análisis dinámico

		AppsPlayground [Rastogi et al. 2013]	CooperDroid [Tam et al. 2015]	TaintDroid [Enck et al. 2014]	DroidScope [Yan and Yin 2012]	MAdFraud [Crussell et al. 2014]	VetDroid [Zhang et al. 2013]	PREC [Ho et al. 2014]	WifiLeaks [Achara et al. 2014]
Input Simulation	User input simulation	●	●	○	○	○	○	●	○
	User input simulation: fuzzing	●	○	○	○	○	○	●	○
	User input: intelligent input generation (e.g., logins, zip codes)	●	●	○	○	○	○	○	○
	Network access simulation	○	○	○	○	○	○	○	○
	System event simulation	●	●	○	○	○	○	○	○
	System event simulation: fuzzing	●	○	○	○	○	○	○	○
	System event simulation: intelligent input generation	○	●	○	○	○	○	○	○
Techniques	Taint tracking	●	○	●	●	○	●	○	○
	Syscall traces	○	●	○	●	○	●	●	○
	Android API layer traces	●	○	○	●	○	○	○	○
	Library call traces	○	○	○	○	○	○	○	○
	Network traces	○	○	○	●	●	○	○	●
	Native instruction traces	○	○	○	●	○	○	○	○
	Dalvik instruction traces	○	○	○	●	○	○	○	○
	Concolic execution	○	○	○	○	○	○	○	○
Control- and DataFlow Tracking	Multiple applications	●	●	●	●	○	○	○	○
	Multiple applications: system applications	●	●	●	●	○	○	○	○
	Multiple applications: user applications	●	●	●	●	○	○	○	○
	Native code	○	●	○	●	○	○	○	○
Resiliency	Emulator detection detection	○	○	N/A	○	○	○	○	○
	Emulator Obfuscation—mimicking realistic environment	●	○	N/A	○	○	○	○	○
	Detects logic bombs or context-sensitive behavior	○	○	○	●	○	○	●	○
Misc	Extensible	○	○	○	○	○	○	○	○
	Useful for any app	●	●	●	●	○	●	●	○
	Publicly available	●	◐	●	●	○	○	○	○
	Benign only/malicious/dual use	D	M	B	M	M	D	M	D
Vulnerabilities Detected	Malicious activity	●	●	○	●	●	●	●	○
	Detect sensitive information leaks	●	◐	●	●	○	●	○	●
	Authentication	○	○	○	○	○	○	○	○
	Cryptography	○	○	○	○	○	○	○	○
	Data validation	○	○	○	○	○	○	○	○
	Intent spoofing	○	○	○	○	○	○	○	○
	Unauthorized intent receipt	○	○	○	○	○	○	○	○
	Configuration and deployment management	○	○	○	○	○	○	○	○
	Permission misuse	○	○	○	○	○	●	○	○
	Plagiarism detection	○	○	○	○	○	○	○	○
	Other	•	○	○	○	•	○	•	○

Seguridad

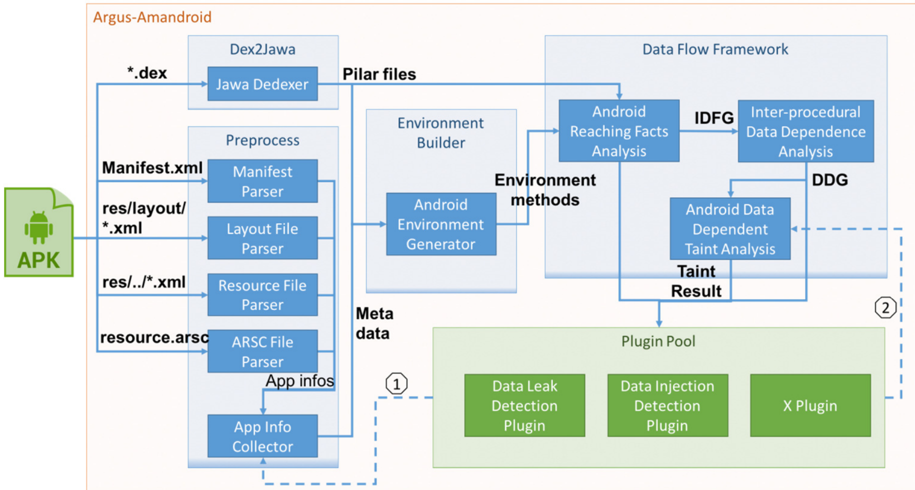
Herramientas de análisis

- Diferentes herramientas permiten analizar la seguridad de aplicaciones Android.
- Su uso por parte de desarrolladores y auditores no siempre es sencillo (puede llegar a ser un desafío).
- En ocasiones, la salida que proporcionan puede ser difícil de interpretar (cuando funcionan)

Seguridad

Herramientas de análisis

Argus SAF [Static Analysis Framework],
antes Amandroid: http://pag.arguslab.org/argus-saf



①: A plugin can control what kind of app information are interesting, and for non-interesting app it can ignore.
②: A plugin can provide it's own "SourceAndSinkManager" to control the source and sink specification and do taint analysis.

Seguridad

Herramientas de análisis

Argus SAF [Static Analysis Framework],
antes Amandroid: http://pag.arguslab.org/argus-saf

Problemas de seguridad detectados (CCS'2014) mediante análisis de flujo de datos [data flow analysis]:

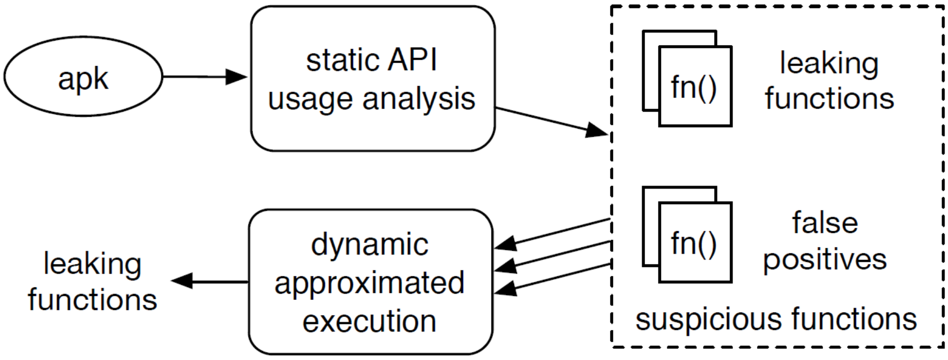
- Data leaks: passwords & OAuth tokens
  (p.ej. entradas en logs, almacenamiento en SharedPreferences o envío a través de TCP/IP)
- Data injection: Intent injection
  ("An intent is an abstract description of an operation to be performed")
- API misuse: Crypto API misuse

Seguridad

Herramientas de análisis

AppAudit

http://appaudit.io/



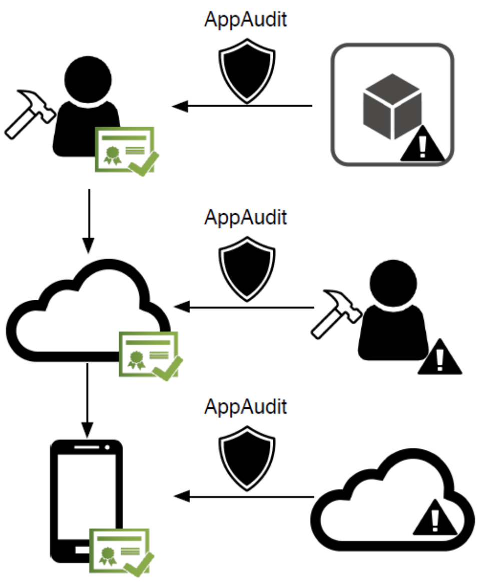
Sensitive data leaks (SP'2015):
hybrid (static/dynamic) taint analysis tool

Seguridad

Herramientas de análisis

AppAudit

http://appaudit.io/



PATDroid
Program Analysis
Toolkit for Android

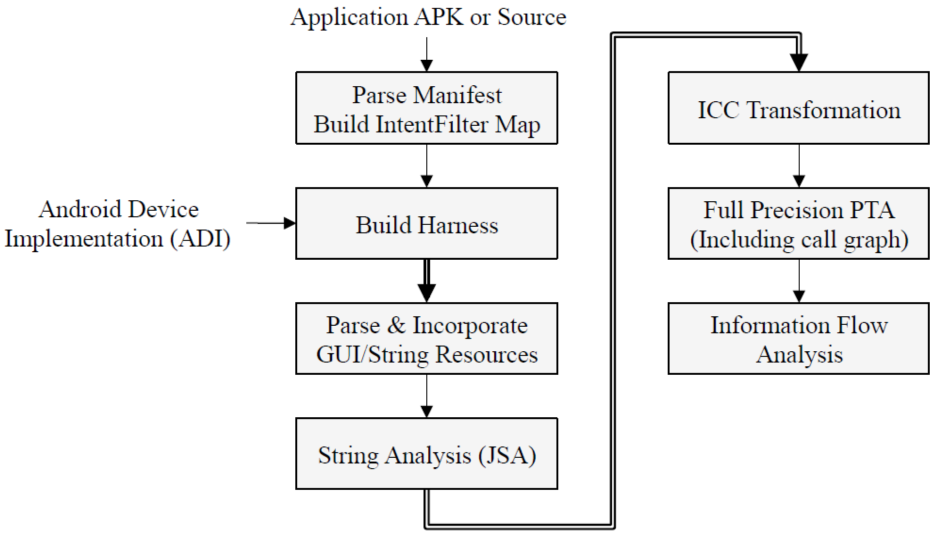
Casos de uso:
producción, distribución e instalación

Seguridad

Herramientas de análisis

DroidSafe

http://mit-pac.github.io/droidsafe-src/



Seguridad

Herramientas de análisis

DroidSafe

http://mit-pac.github.io/droidsafe-src/

Soot

Java optimization Framework

(CASCON'1999)

Análisis de flujos de datos maliciosos en el código fuente de aplicaciones Android y en ficheros APK.

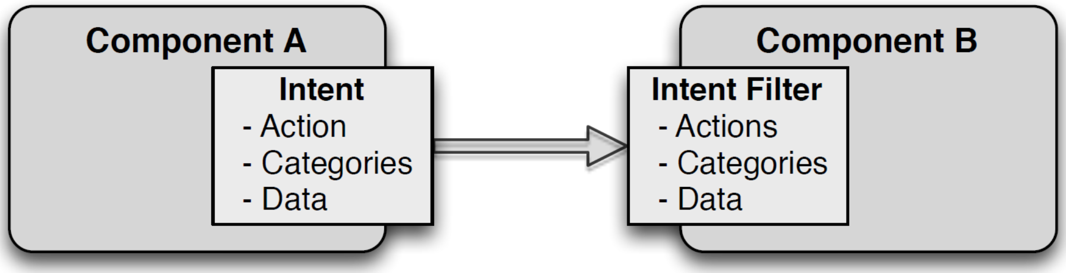
Un auditor tardó ¡15 horas en instalarla!

Seguridad

Herramientas de análisis

IC3: Inter-Component Communication Analysis for Android

http://siis.cse.psu.edu/ic3/index.html (ICSE'2015)



Versión anterior: **Epicc** (USENIX'2013)

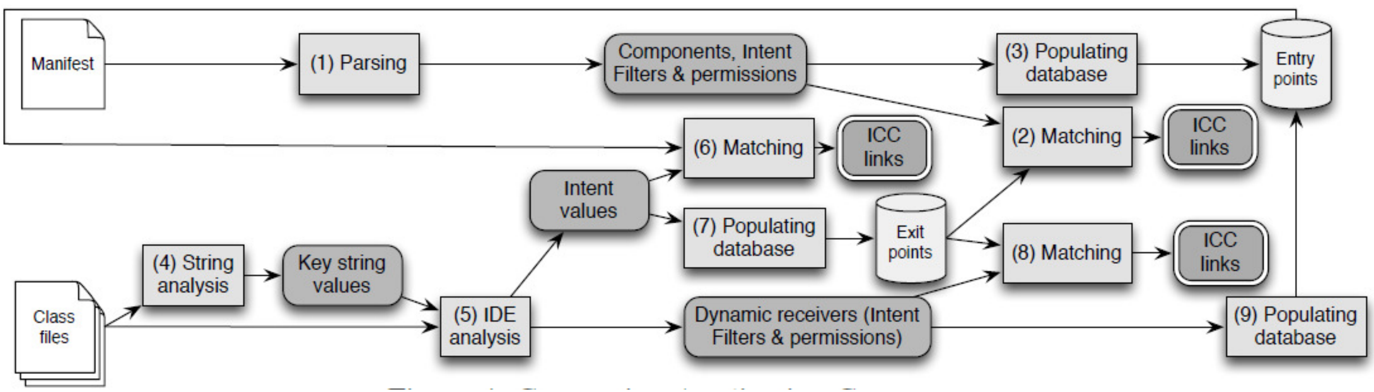
Effective and Precise ICC [InterComponent Communication]

http://siis.cse.psu.edu/epicc/

Seguridad

Herramientas de análisis

Epicc (USENIX'2013)



ICC as an instance of the Interprocedural Distributive Environment (IDE) data flow problem.

Seguridad

Herramientas de análisis

Epicc (USENIX’2013)

Capaz de detectar 7 tipos de vulnerabilidades:

- activity hijacking,
- broadcast theft (sniffing),
- malicious broadcast injection,
- malicious activity launch,
- protected system broadcast without action check,
- malicious service launch, and
- service hijacking.

Seguridad

Herramientas de análisis

FlowDroid (PLDI’2014)

https://blogs.uni-paderborn.de/sse/tools/flowdroid/

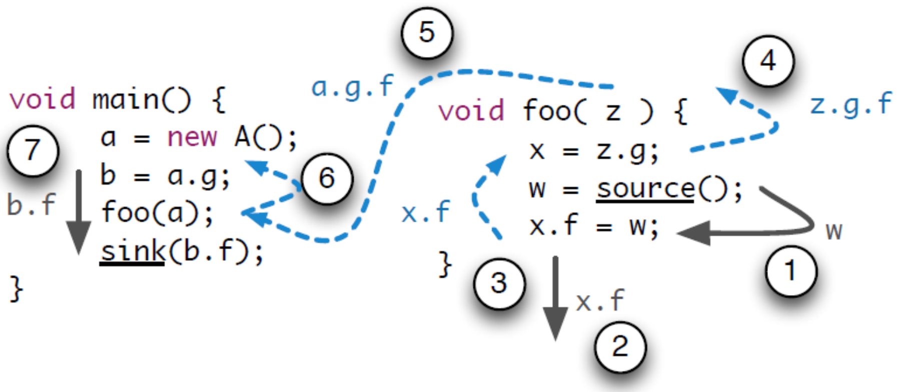
Static taint analysis tool:
context, flow, field, object-sensitive,
and lifecycle-aware taint analysis.

Detecta un solo tipo de vulnerabilidad
[information leakage] analizando conexiones.

Herramientas de análisis

FlowDroid (PLDI'2014)

https://blogs.uni-paderborn.de/sse/tools/flowdroid/



Herramientas de análisis

MalloDroid

https://github.com/sfahl/mallodroid

Vulnerabilidad (CCS'2012):

Detect improper TLS certificate validation
that may allow Man-in-the-Middle (MitM) attacks.

Script basado en Androguard

https://github.com/androguard/androguard

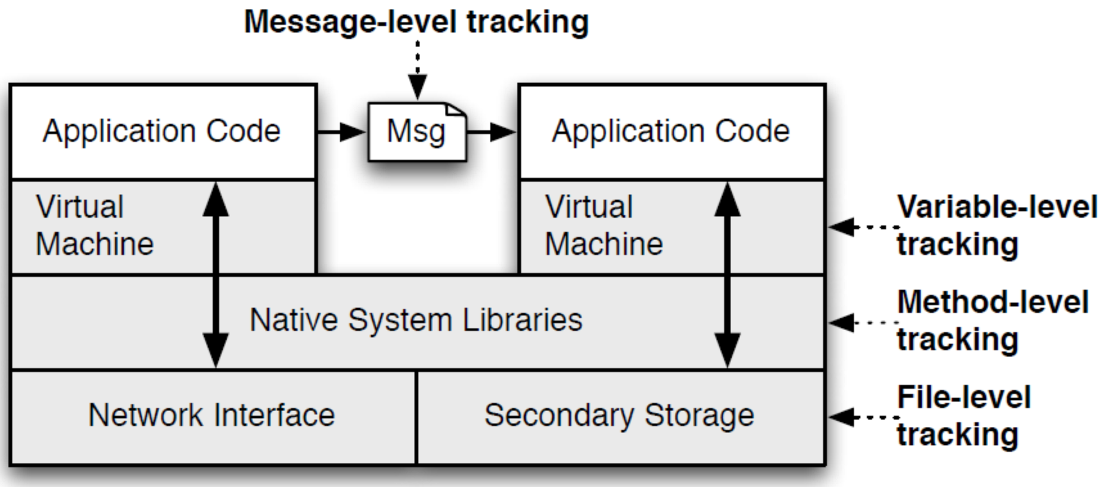
"Reverse engineering, Malware analysis of Android applications ... and more (ninja) !"

Seguridad

Herramientas de análisis

TaintDroid (OSDI'2010)

http://www.appanalysis.org/

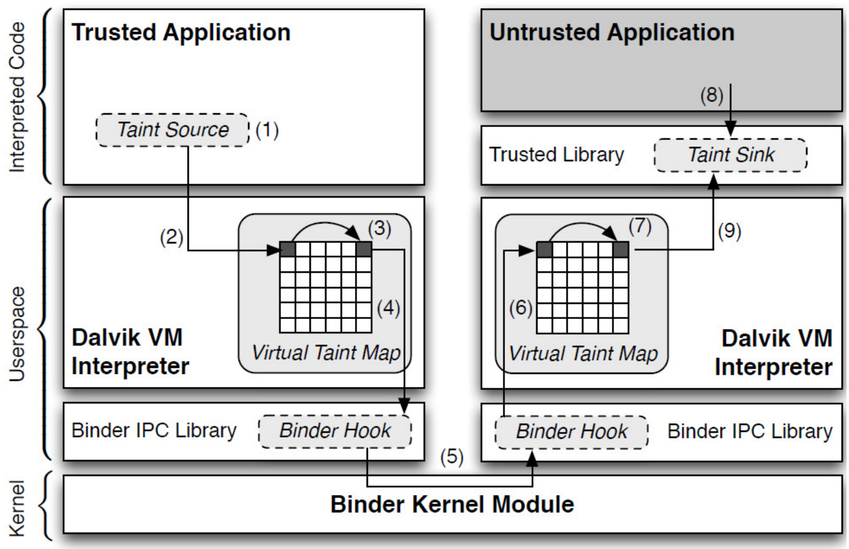


Seguridad

Herramientas de análisis

TaintDroid (OSDI'2010)

http://www.appanalysis.org/



Seguridad

Herramientas de análisis

TaintDroid (OSDI'2010)

http://www.appanalysis.org/

Dynamic taint analysis tool designed to analyze commonly used applications and identify leaks of privacy-sensitive information.

Designed to run on a real device…

requires the user to download and build the Android AOSP project (https://source.android.com/).

Seguridad

Herramientas de análisis: Data flow

Analysis	Projects	Description	Bypassable?
Dynamic	TaintDroid	Dynamic tracking on Dalvik VM	
	NDroid	Dynamic tracking on native code (JNI)	
	Capper	Tracking with Dalvik bytecode rewriting	Taint cleanse through control dependence or side channels
	VetDroid	Permission-based discovery of taint sources and sinks	
	SuSi	Machine learning for discovery of taint sources and sinks	
Static	CHEX	Intercomponent (ICC) awareness	
	Epicc	Reduce ICC to IDE problem	
	FlowDroid	Lifecycle awareness	
	Amandroid	ICC & lifecycle awareness	
	IccTA	ICC & lifecycle awareness	Code encryption, Java reflection, dynamic code loading, etc.
	EdgeMiner	Model implicit control flow through Android framework	
	DroidSafe	Model Android specific features with "stubs"	
Hybrid	AppIntent	Event-constrained symbolic exec	
	SmartDroid	Directional dynamic execution	
	Intellidroid	Event-focused API reachability analysis	Any technique in static or dynamic.
	Harvester	Execution on sliced app components	

Herramientas de análisis

- Sensitive data flows detected by tools are not necessarily suspicious or malicious, as most of them are actually necessary to the apps' functionalities and should be allowed.
- Judging the legitimacy of detected privacy disclosures usually requires domain knowledge, and thus is hard to be automated.

Herramientas de análisis: Malicious behavior

Detection	Projects	Description	Bypassable?
Execution	DroidScope/CopperDroid NJAS/Boxify	Execute/monitor apps in an emulator Sandbox interaction between apps and Android/OS	Emulator detection techniques Sandbox escape
Model Checking	Pegasus AppContext	Permission Event Graph based checking Check security-sensitive behavior contexts	Code encryption and reflection
WYSIWYX	WHYPER/AutoCog CHABADA/ACODE AsDroid	Compare permissions to app description Compare API calls to app description Evaluate app behaviors based on UI texts	Ambiguous/fake description Dynamic UI/code loading
Machine Learning	Hao et al. DroidAPIMiner Drebin DroidSIFT MUDFLOW	Check permission requests and app categories Check API calls, packages, and parameters Check permissions, APIs, and network activity Check contextual API dependency graph Check sensitive data accesses and usage	Feature manipulation

p.ej.

- Monitorización de secuencias de llamadas al sistema
- WYSIWYX [What You See is What You Execute]

Seguridad

Prácticas de seguridad en el ecosistema Android

Toward Engineering a Secure Android Ecosystem: A Survey of Existing Techniques

MENG XU, CHENGYU SONG, YANG JI, MING-WEI SHIH, KANGJIE LU, CONG ZHENG, RUIAN DUAN, YEONGJIN JANG, BYOUNGYOUNG LEE, CHENXIONG QIAN, SANGHO LEE, and TAESOO KIM, Georgia Institute of Technology

The openness and extensibility of Android have made it a popular platform for mobile devices and a strong candidate to drive the Internet-of-Things. Unfortunately, these properties also leave Android vulnerable, attracting attacks for profit or fun. To mitigate these threats, numerous issue-specific solutions have been proposed. With the increasing number and complexity of security problems and solutions, we believe this is the right moment to step back and systematically re-evaluate the Android security architecture and security practices in the ecosystem. We organize the most recent security research on the Android platform into two categories: the software stack and the ecosystem. For each category, we provide a comprehensive narrative of the problem space, highlight the limitations of the proposed solutions, and identify open problems for future research. Based on our collection of knowledge, we envision a blueprint for engineering a secure, next-generation Android ecosystem.

CCS Concepts: • **Security and privacy → Mobile platform security**; *Malware and its mitigation*; Social aspects of security and privacy

"Toward Engineering a Secure Android Ecosystem: A Survey of Existing Techniques", ACM Computing Surveys, November 2016

Seguridad

Prácticas de seguridad en el ecosistema Android

- **Malicious behavior detection:** Google Play introduced *Bouncer* [NDSS'2014], a malware scanning service to detect malicious in-store and prestore apps.
- **Repackaging detection and prevention**
- **Infection channel cut-off** (cut off malware distribution from app stores and untrusted sources).
- **Incentive elimination:** Android's profit model involves sharing revenue between developers and the app store (e.g., Google Play and Amazon App Store). Such a model indirectly undermines illegal malware by suppressing malware writers' incentives.

Seguridad

Vulnerabilidades más habituales

Escalado de privilegios [system privilege escalation]

Superficie de ataque:

- Kernel Linux
- Native daemons
- Third-party drivers

Otra fuente de problemas de seguridad de este tipo: rootkits / jailbreaking

Seguridad

Vulnerabilidades más habituales

Modelo de permisos

- Asignación de permisos (desarrolladores de apps).
- Problema de usabilidad: el usuario tiene que aceptar todos los permisos solicitados por una app si desea utilizarla (y la app puede luego abusar de ellos).

Recomendación: **PoLP [Principle of Least Privilege]**

Versiones alternativas de Android: p.ej. Cyanogenmod & Blackphone

Seguridad

Permisos en algunas apps populares (Apple Store) 2021

LOS "DATOS VINCULADOS CONTIGO" QUE USAN LAS 'APP'

Esta es la información que las aplicaciones pueden recopilar y vincular a la identidad del usuario, según se indica en el Apple Store.

	Datos de contacto	Identificadores	Contenido de usuario	Ubicación	Datos de uso	Diagnósticos	Contactos	Historial de búsqueda	Compras	Información financiera	Otros datos	Historial de navegación	Datos sensibles	Salud y forma física
Facebook y Messenger	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Instagram	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Amazon	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓
Google	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓		
TikTok	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓		
Uber	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓		✓	
YouTube	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓		
Twitter	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓		
Tinder	✓	✓	✓	✓	✓	✓	✓		✓				✓	
McDonald's	✓	✓	✓	✓	✓	✓		✓	✓	✓				
WhatsApp	✓	✓	✓	✓	✓		✓		✓	✓				
Meet	✓	✓	✓	✓	✓	✓		✓			✓			
Candy Crush	✓	✓	✓	✓	✓	✓	✓		✓					
Teams	✓	✓	✓	✓	✓	✓	✓							
YouTube Kids	✓	✓	✓	✓	✓	✓		✓						
Clubhouse	✓	✓	✓		✓	✓	✓							
Zoom	✓	✓	✓	✓	✓	✓								
Telegram	✓	✓					✓							
Lego	✓	✓	✓											
EMT Madrid	✓			✓										
Signal	✓													

Expansión

Fuente: Elaboración propia con datos de las 'app'

Seguridad

Vulnerabilidades más habituales

Modelo de permisos: Transitividad vía Intents (ICC)

- **Developers** should write robust and secure interfaces that only accept intents from apps with required permissions. Failure to do so results in **confused deputy attack** where the deputy app fails to check whether the calling app has the credentials to use their permission-protected interfaces.
- For **users**, permission transitivity makes it harder to foresee how an app might use permissions from other apps, which enables **collusion attack** where two or more malicious apps with distinct but limited permissions collaborate to effectively generate a joint set of permissions.

Seguridad

Vulnerabilidades más habituales

Modelo de permisos: Posibles mejoras

- Comprobación [estática] de solicitudes de permisos, p.ej. Stowaway, PScout, WHYPER, AutoCog
- Políticas de privacidad (intercepción ICC), p.ej. Apex, CRePE, AppFence
- Descomposición de permisos, p.ej. Constroid, Aurasium, Dr. Android & Mr. Hyde

  EJEMPLO: Acceso a Internet por dominios (como en las extensiones y Apps de Chrome) en vez de un permiso genérico INTERNET.

Seguridad

Vulnerabilidades más habituales

Modelo de permisos: Posibles mejoras

- Descomposición de la asignación de permisos (a nivel de componentes en vez de a nivel de apps) p.ej. Brahmastra, Compac, FlexDroid
- ICC Tracing (análisis dinámico), p.ej. Saint, FlaskDroid, QUIRE, IPC Inspection
- Control de flujo de información, [DIFC: Decentralized Information Flow Control] p.ej. Aquifer, Maxoid…

Seguridad

Vulnerabilidades más habituales

Modelo de permisos: Posibles mejoras

Type	Solution	Description	Required Modification
①	Stowaway	Extract permission-to-API map through API testing.	-
	PScout	Extract permission-to-API map from Android source code.	-
	WHYPER	Ensure description-to-permission fidelity.	-
	AutoCog	Ensure description-to-permission fidelity.	-
②	Apex	Intercept sensitive API calls and filter against predefined policies.	App Installer, ICC Monitor
	CRePE	Intercept sensitive API calls and filter against predefined policies.	ICC Monitor
	AppFence	Supply sensitive API calls with mock data/reject network transmission.	Dalvik VM, Resourec Manager
③	Constroid	Data-centric access control by policies.	ContentProvier, ICC Monitor
	Dr. Android	Sensitive API drop-in replacement.	App Repackaging
	Aurasium	In-app `libc` interposition.	App Repackaging
④	AdDroid	Run advertisements as system services.	System Service, Libc
	AdSplit	Split ad library and app into separate processes.	App Repackaging
	LayerCake	Provide in-app privilege separation through process.	View Object
	Compac	Provide component-level permission assignment.	ICC Monitor, Kernel
	FlexDroid	Provide in-app privilege separation through call stack tracing	Kernel, Dalvik VM, Libc
⑤	IPC Inspection	Permission check across the whole API access call chain.	ICC Monitor
	SORBET	Permission check across the whole API access call chain.	ICC Monitor, Kernel
	XManDroid	Permission check across the whole API access call chain.	App Installer, ICC Monitor
	Quire	Permission check across the whole API access call chain.	ICC Monitor, Kernel
	Saint	Allow developers to define policies on interface access.	App Installer, ICC Monitor
⑥	Aquifer	Allow developers to attach policies on data shared.	ICC Monitor, Kernel
	Jia et al.	Allow developers to attach policies on data shared.	ICC Monitor, Kernel
	Maxoid	Allow developers to attach policies on data shared.	ICC Monitor, Kernel

Seguridad

Vulnerabilidades más frecuentes

"Abuso" de determinadas características de Android

- Carga dinámica y generación dinámica de código, p.ej. actualizaciones sin pasar por Google Play Store.
- Código nativo vía JNI [Java Native Interface].
- Navegador web [WebView]: interfaz Java-JavaScript.
- Tecnologías "asistivas" (accesibilidad): control de E/S.
- Soporte multiusuario (Android >4.2), p.ej. procesos que siguen ejecutándose tras cambiar de usuario
- ART [Android RunTime] como sustituto de Dalvik VM (Android >5.0): AOT [ahead-of-time] compilation = ejecución de código nativo…

Seguridad

Vulnerabilidades más frecuentes

Reempaquetado de aplicaciones

86% del malware
distribuido como versiones de apps conocidas,
p.ej. Pokémon Go o WhatsApp

Seguridad

Vulnerabilidades más frecuentes

Reempaquetado de aplicaciones

TÉCNICAS DE DETECCIÓN:

Solution	App representation	Similarity comparison
DroidMOSS	Hash of opcodes block	Edit distance
DNADroid	Data dependence graph	Subgraph isomorphism
AnDarwin	Data dependence graph	Subgraph isomorphism
Androguard	Regex string of CFG	Normalized compression distance
PiggyApp	Regex string of CFG	Normalized compression distance
Centroid	Centroid of CFG	Method centroid distance
DroidSim	Component-based CFG	Jaccard coefficient
Juxtapp	Feature hashing	Jaccard similarity metric
ViewDroid	View-event graph	Subgraph isomorphism
ResDroid	View-event graph and statistics	Hierarchical Clustering

Seguridad

Vulnerabilidades más frecuentes

Reempaquetado de aplicaciones

TÉCNICAS DE PREVENCIÓN:

- Marcas de agua, p.ej. AppInk
- Verificación de autenticidad usando un dominio propio (vía DNS), no sólo al firmar la aplicación.
- Ofuscación de bytecodes (dificulta el trabajo de ingeniería inversa necesario para el reempaquetado), p.ej. DIVILAR

Caso práctico: WhatsApp

SIN CLASIFICAR

ccn-cert
centro criptológico nacional

Informe de Amenazas
CCN-CERT IA-21/16

Riesgos de uso de WhatsApp

Seguridad

Caso práctico

WhatsApp

Informe de Amenazas CCN-CERT IA-21/16

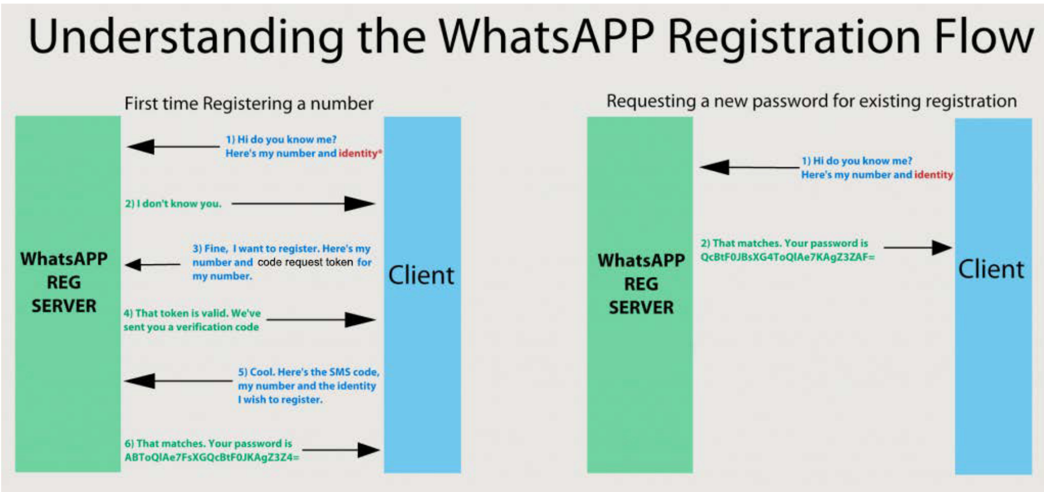
Riesgos de uso de WhatsApp

Septiembre de 2016

Seguridad

Caso práctico: WhatsApp

Seguridad en el proceso de alta y verificación de usuarios



- Un intruso podría hacerse con la cuenta de usuario de WhatsApp de otra persona, leer los mensajes que reciba e incluso enviar mensajes en su nombre.

Seguridad

Caso práctico: WhatsApp

Fallos en los protocolos de red, p.ej. SS7

- Secuestro de cuentas, tanto de WhatsApp como de otras aplicaciones como Telegram, utilizando fallos conocidos en el protocolo de telecomunicaciones **SS7** (Signalling System No. 7).
- Un intruso puede hacerse con la cuenta de usuario de WhatsApp de otra persona, leer los mensajes que reciba e incluso enviar mensajes en su nombre.

Seguridad

Caso práctico: WhatsApp

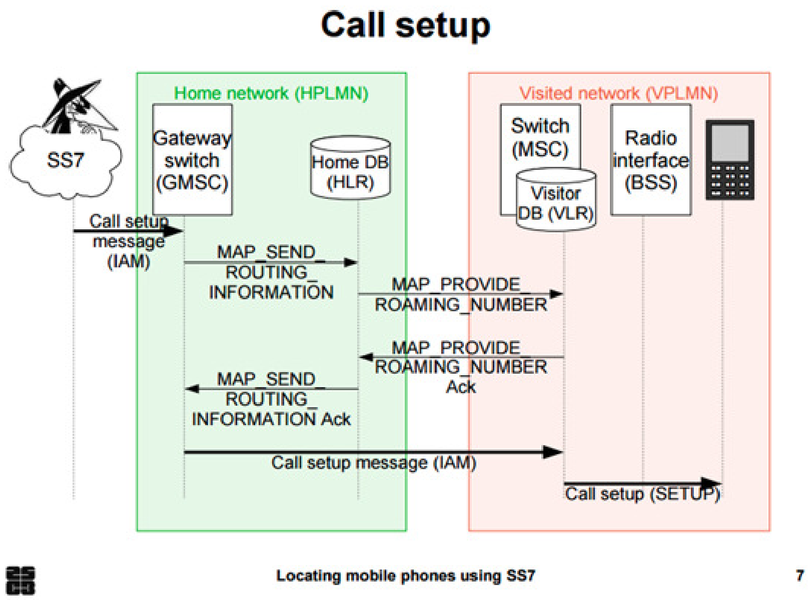
Fallos en los protocolos de red, p.ej. SS7

- Protocolo diseñado por AT&T en 1975: intercambio de información sobre una red digital para efectuar el enrutamiento, establecimiento y control de llamadas.
- Forma parte, entre otros, del funcionamiento interno de servicios como los SMS. En el caso de que un atacante consiguiera acceso al sistema SS7, podría interceptar o grabar llamadas, leer SMS, o detectar la localización del dispositivo utilizando el mismo sistema que la red del teléfono.

Seguridad

Caso práctico: WhatsApp

Fallos en los protocolos de red, p.ej. SS7



Seguridad

Caso práctico: WhatsApp

Fallos en los protocolos de red, p.ej. SS7

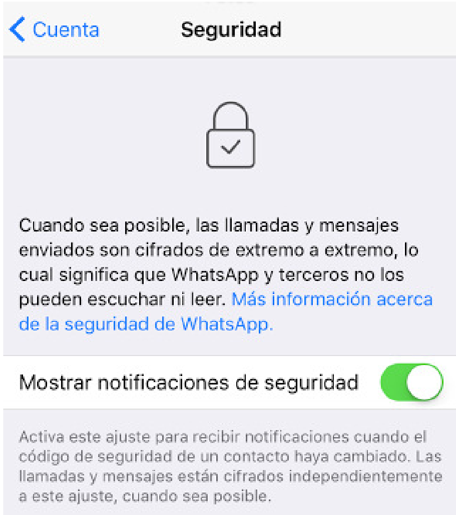
- El ataque se realiza de forma sencilla, haciendo creer a la red telefónica que el teléfono del atacante tiene el mismo número que la víctima.

- De esta forma, se consigue recibir un código de verificación de WhatsApp válido, teniendo acceso completo a la cuenta de la víctima, independientemente del cifrado incluido en las comunicaciones.

Seguridad

Caso práctico: WhatsApp

Fallos en los protocolos de red, p.ej. SS7



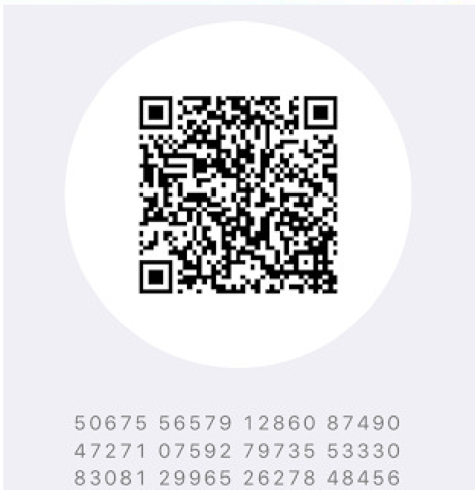
- Al tratarse de un fallo de red,
  no dependiente de la aplicación:
  no existe una forma directa
  de resolver este fallo de seguridad.

- Recomendación:
  activar la opción de
  "Mostrar notificaciones de seguridad"

Seguridad

Caso práctico: WhatsApp

Códigos de seguridad de WhatsApp



- Cada chat tiene un código de seguridad único que es usado para confirmar que las llamadas y mensajes que se envían a ese chat, están cifrados de extremo a extremo. Este código se encuentra en la pantalla de información de los contactos y está disponible en forma de código QR y de 60 dígitos.

Seguridad

Caso práctico: WhatsApp

Borrado inseguro de comunicaciones

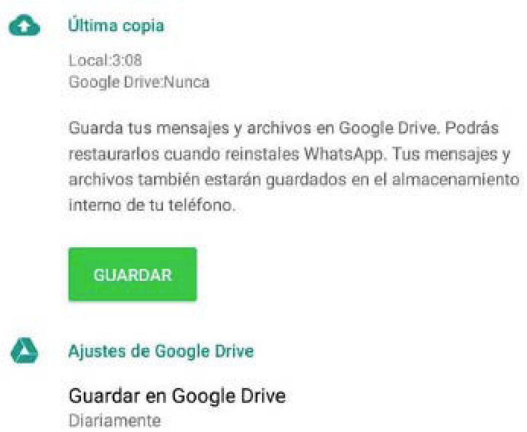
- Fallo que se utiliza para obtener los registros de las conversaciones utilizando técnicas forenses.
- Origen: Base de datos SQLite.
- Motivo: El proceso de borrado de una conversación, mensaje o grupo es sencillo en el teléfono, pero no implica la eliminación directa de los mensajes, sino que estos quedan marcados como libres, de tal forma que puedan ser sobrescritos por nuevas conversaciones o datos cuando sea necesario.

Seguridad

Caso práctico: WhatsApp

Borrado inseguro de comunicaciones

- Otro posible origen: Opción de copia de seguridad usando iCloud en iOS o Google Drive en Android.



Seguridad

Caso práctico: WhatsApp

Difusión de información sensible

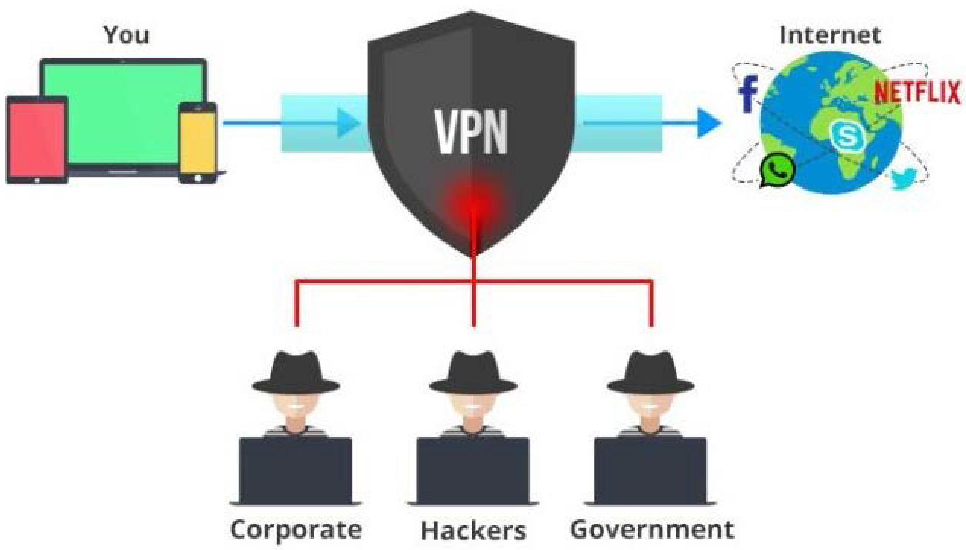
Durante la conexión inicial, en texto plano, como ASCII (originalmente) o en binario (desde que se usa cifrado E2E):

- Sistema operativo del cliente.
- Versión de la aplicación en uso.
- Número de teléfono registrado.

Seguridad

Caso práctico: WhatsApp

Difusión de información sensible

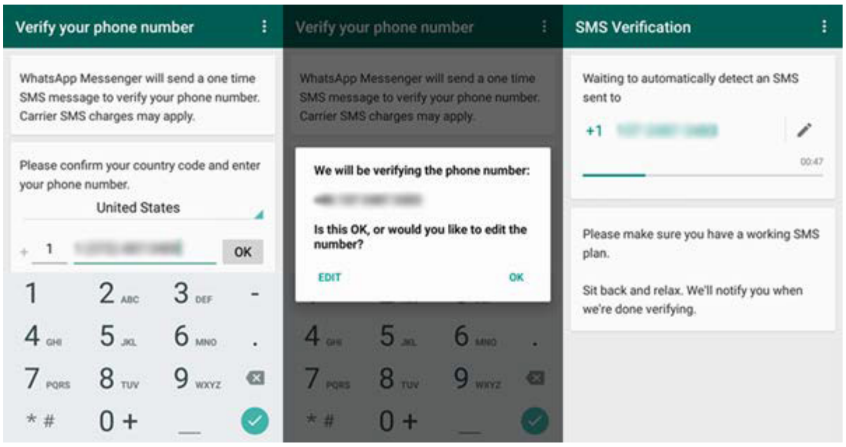


Uso de conexiones VPN

Seguridad

Caso práctico: WhatsApp

Robo de cuentas mediante SMS y acceso físico



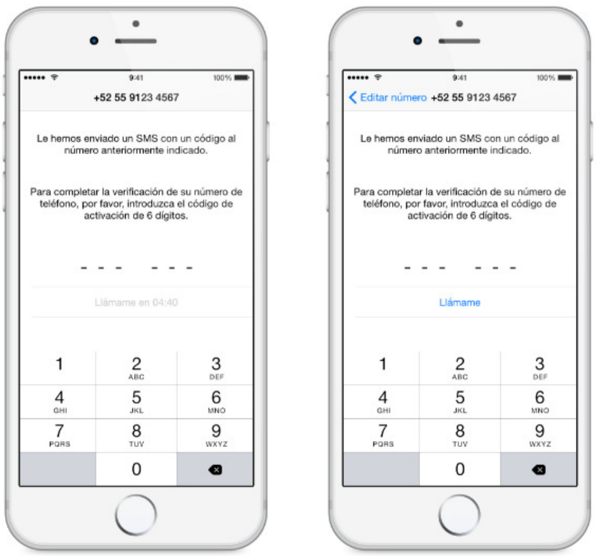
Recomendación: Desactivar la previsualización del remitente y contenido de los mensajes SMS en la pantalla de bloqueo del terminal.

Seguridad

Caso práctico: WhatsApp

Robo de cuentas mediante llamada y acceso físico

No existe una opción, tanto para **Android** como para **iPhone**, que fuerce al usuario a desbloquear el terminal para poder responder a una llamada, por lo que un atacante con acceso físico siempre podrá responder y completar el ataque.



Seguridad

Caso práctico: WhatsApp

Phishing usando WhatsApp Web

https://web.whatsapp.com

WhatsApp

Usa WhatsApp en tu teléfono para escanear el código

☑ Mantener sesión iniciada

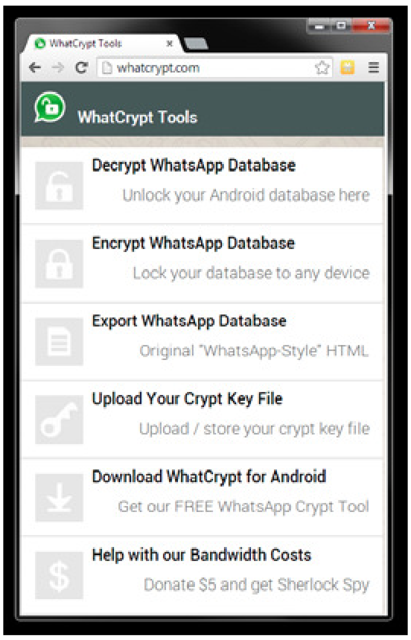
Para reducir el consumo de datos móviles, conecta tu teléfono a una red Wi-Fi

p.ej. Código QR para robar las credenciales de inicio de sesión.

Seguridad

Caso práctico: WhatsApp

Datos "cifrados" almacenados en SQLite



Seguridad

Caso práctico: WhatsApp

Falta de cifrado en versiones antiguas

- Acceso a los números de teléfono y al contenido de los mensajes. Si el GPS estaba activado, también a la ubicación del usuario.
- Evolución: Sin cifrado ➔ contraseña basada en IMEI o dirección MAC de la tarjeta Wi-Fi ➔ uso inseguro del algoritmo RC4 ➔ cifrado extremo a extremo (E2E)

Seguridad

Caso práctico: WhatsApp

Intercambio de datos entre WhatsApp y Facebook

Compra de WhatsApp por Facebook, febrero de 2014:

"*El respeto a su privacidad está codificado en nuestro ADN, y hemos construido WhatsApp en torno al objetivo de conocer tan poco acerca de usted como sea posible.*"

25 de agosto de 2016:

Modificación de términos y condiciones de uso :-(

Seguridad

Caso práctico: WhatsApp

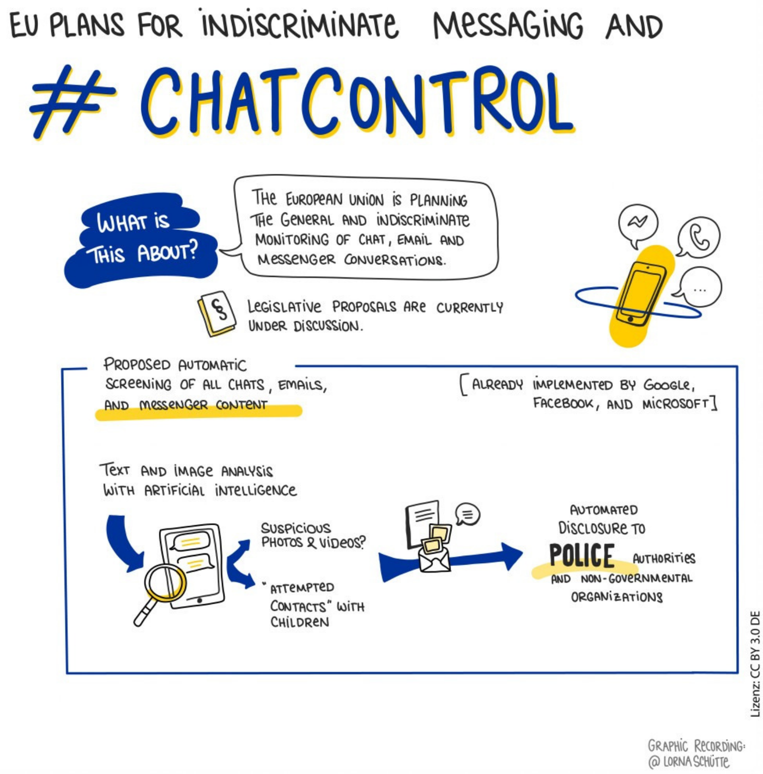
Versiones fraudulentas en Google Play Store

https://thehackernews.com/2017/11/fake-whatsapp-android.html

Noviembre de 2017 :-(

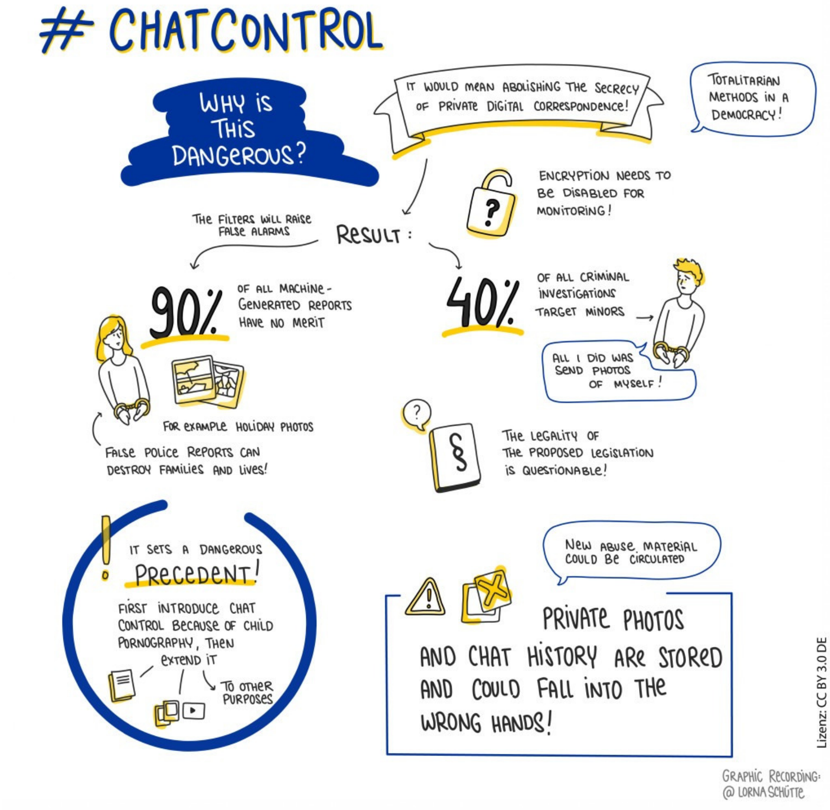
Seguridad

#ChatControl



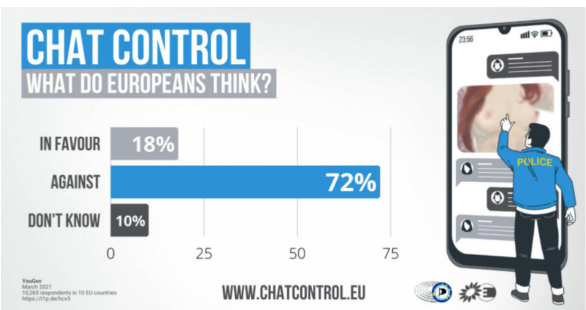
Seguridad

#ChatControl



Seguridad

#ChatControl



NOTICIAS

ChatControl, ¿un paso atrás para la privacidad en la Unión Europea?

Publicado el 9 julio, 2021 por Juan Ranchal

La Comisión Europea ya ha anunciado un reglamento de seguimiento para que este control de los chat sea **obligatorio para todos los proveedores de correo electrónico y mensajería**. Los servicios de mensajería cifrados de extremo a extremo, incluso los considerados tan seguros y privados como Signal, se verían obligados a instalar una puerta trasera.

https://www.muycomputer.com/2021/07/09/chatcontrol-paso-atras-privacidad/

Seguridad

¿Cómo detectar si el teléfono está infectado?

Muy complicado, salvo que el "malware" genere señales visibles…

- Desgaste prematuro de la batería.
- Ralentización de operaciones habituales.

Seguridad

Recomendaciones generales

- Mantener el teléfono bloqueado.
- Eliminar las previsualizaciones de los mensajes.
- Cuidado con las solicitudes de permisos de las apps.
- Desactivar la conectividad adicional del teléfono cuando no se vaya a utilizar, p.ej. Wi-Fi o Bluetooth.
- Restringir el acceso físico al dispositivo.

Seguridad

Android covert channels

Formas de enviar información saltándose los controles…

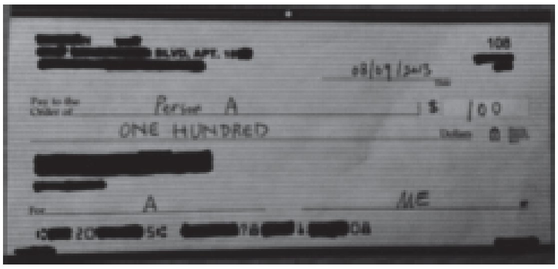
- Vibración
- Brillo de la pantalla
- Volumen
- Ultrasonidos
- Cerrojos de ficheros
- Tipo de Intent
- Sockets UNIX
- Enumeración de hebras
- Espacio libre en "disco"

…

Seguridad

Android side channels

[shared memory]



"Researchers find way to hack
Gmail with 92 percent success rate"

http://www.cnet.com/news/researchers-find-way-to-hack-gmail-with-92-percent-success-rate/

Qi Alfred Chen, Zhiyun Qian & Z. Morley Mao:
"Peeking into Your App without Actually Seeing It:
UI State Inference and Novel Android Attacks"
Proceedings of the 23rd USENIX Security Symposium,
San Diego, CA, August 2014.

Seguridad

Android side channels

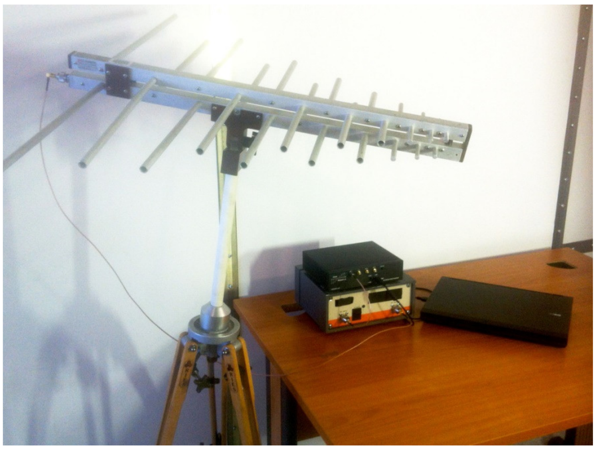
Hardware & software compartido

Type	Channel	Target	Attacks	Mitigation
HW	Accelerometer	Screen taps, PIN	Sensor based	Finer-grained permission Reducing sampling frequency
HW	CPU Cache	Crypto key	Crypto key recovery	Side-channel resistant crypto algorithm Crypto co-processor w/ AES instruction
HW	CPU Cache	Kernel address map	Breaking kernel ASLR	Normalizing page fault handling time Isolation of user and kernel cache use Disabling high-precision timer (e.g., rdtsc)
SW	/proc file system	Illegal user data retrieval	UI State inference	Access restriction to /proc file system Adding indicator on sensitive screen page

p.ej. TapPrint (sensores de movimiento para inferir lo que se tecleaa) o Soundcomber (extracción de información sensible de conversaciones)

Seguridad

… use radio waves to silently trigger voice commands on any Android phone or iPhone that has Google Now or Siri enabled, if it also has a pair of headphones with a microphone plugged into its jack…



… [use] those headphones' cord as an antenna, exploiting its wire to convert surreptitious electromagnetic waves into electrical signals that appear to the phone's operating system to be audio coming from the user's microphone…

"Hackers Can Silently Control Siri From 16 Feet Away"

Wired, October 2015

http://www.wired.com/2015/10/this-radio-trick-silently-hacks-siri-from-16-feet-away/

Seguridad

Sistemas basados en redes neuronales

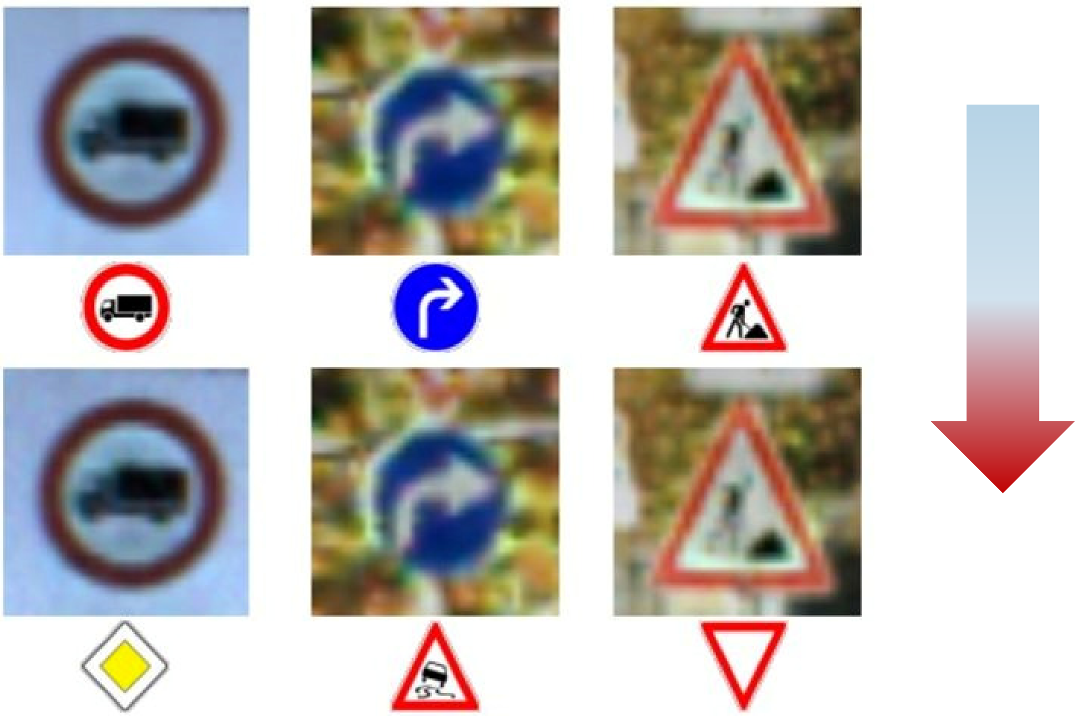
"Two groups, one at Berkeley University and another at Georgetown University, have successfully developed algorithms that can issue speech commands for digital personal assistants, like Siri and Google Now, in the form of bursts of sound unrecognizable to human ears. To a human, these commands just sound like random white noise, but they could be used to tell a voice-activated assistant like Amazon's Alexa to do things that its owner never intended."

Fooling The Machine
The Byzantine Science of Deceiving Artificial Intelligence
Popular Science, 2016
http://www.popsci.com/byzantine-science-deceiving-artificial-intelligence

Seguridad

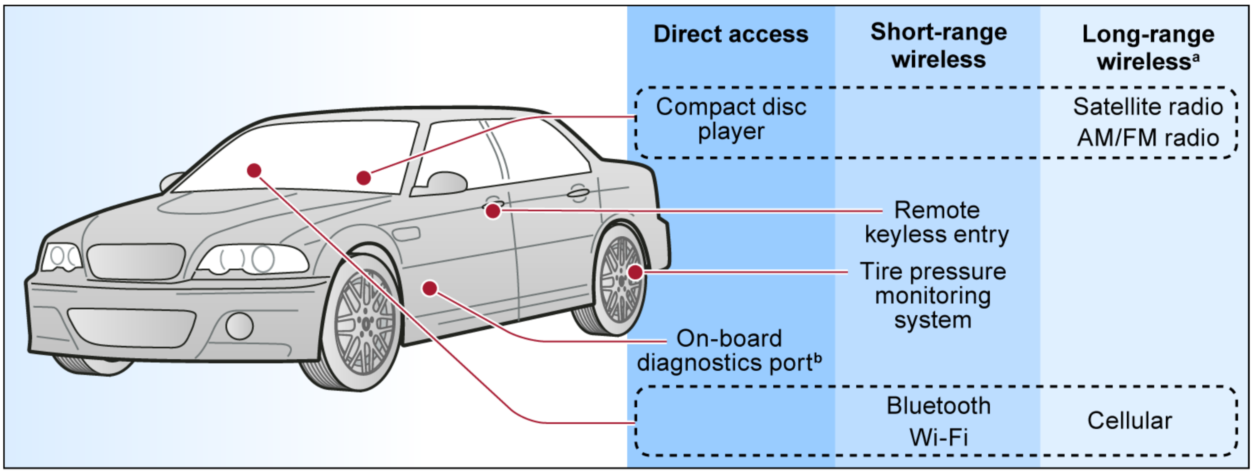
Sistemas basados en redes neuronales

Se les puede engañar a propósito…



Seguridad

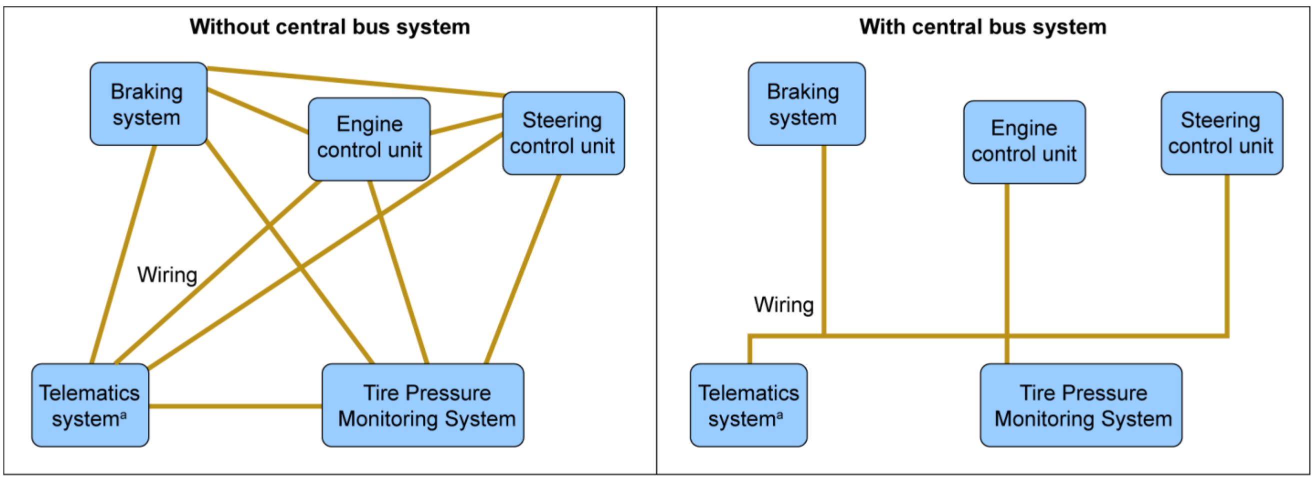
Vehículos: Interfaces



Source: GAO analysis of stakeholder interviews and Checkoway et al, 2011. | GAO-16-350

Seguridad

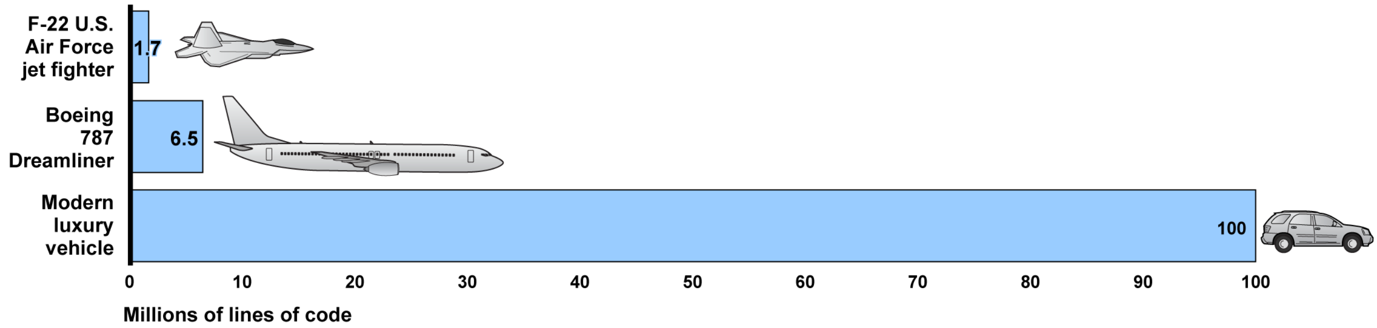
Vehículos: Red de comunicaciones



Source: National Instruments. | GAO-16-350

Seguridad

Vehículos: Software

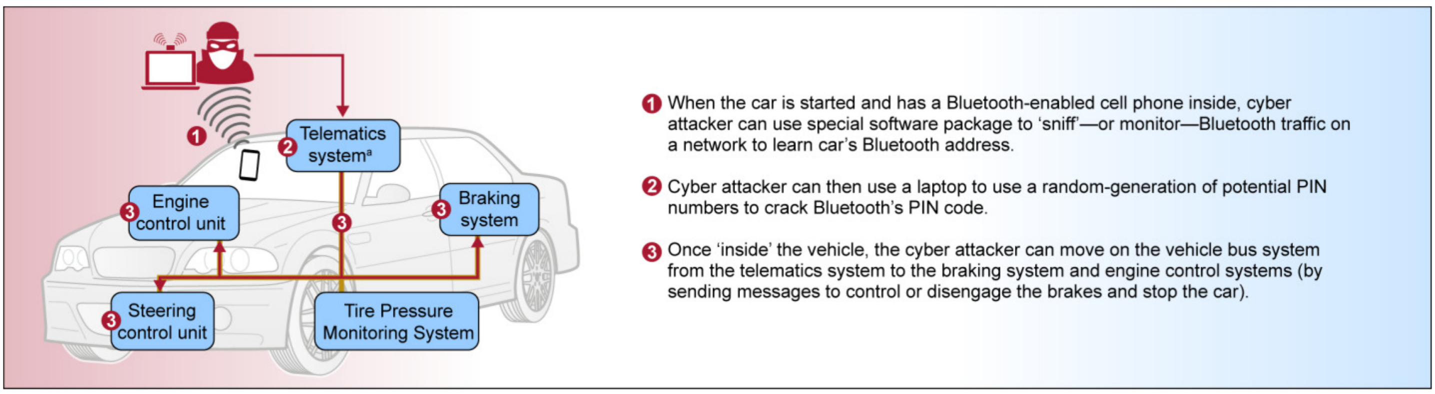


Source: Battelle. | GAO-16-350

Líneas de código (MLOC)

Seguridad

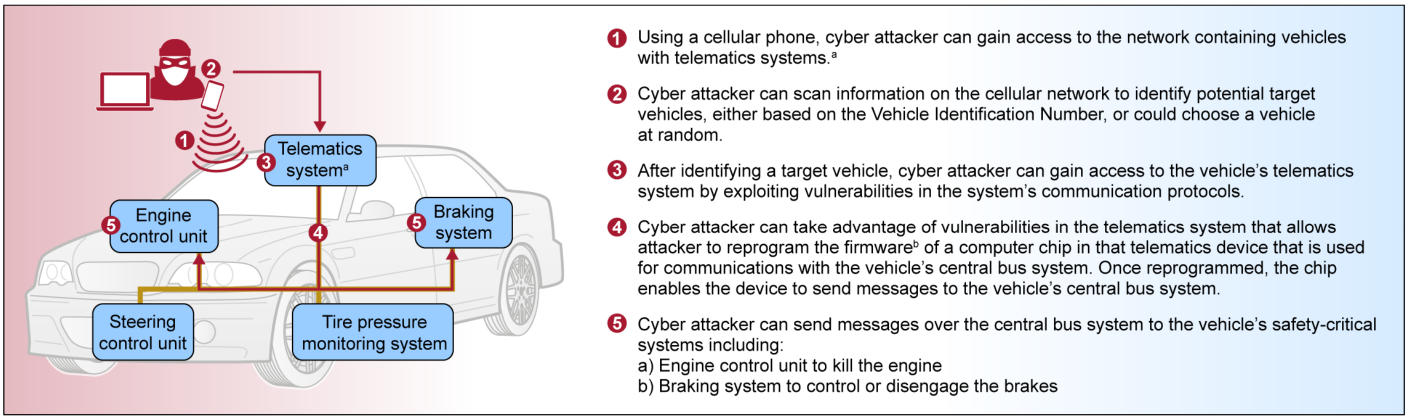
Vehículos: Posible ataque (Bluetooth)



Source: GAO analysis of Checkoway et al, 2011. | GAO-16-350

Seguridad

Vehículos: Posible ataque (teléfono móvil)



Source: GAO analysis of Miller and Valasek, 2015. | GAO-16-350

e.g. MirrorLink (WOOT'2016)
A Security Analysis of an In-Vehicle Infotainment and App Platform
https://www.usenix.org/conference/woot16/

Seguridad

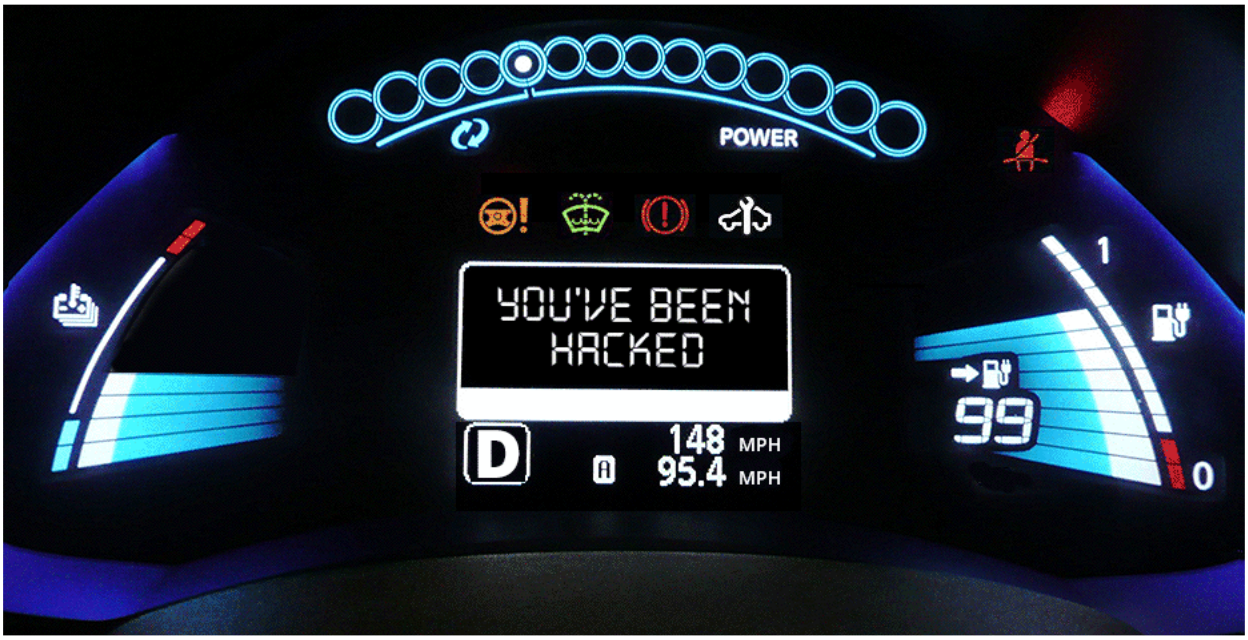
Vehículos: Tesla Model S

"The researchers were able to remotely control the braking system, sunroof, door locks, trunk, side-view mirrors and more"
… through the car's controller area network (CAN) bus.

https://youtu.be/c1XyhReNcHY

Seguridad

Vehículos



Seguridad: Drones

Algunos ejemplos

Johns Hopkins computer security team, June 2016

CPU overload -> shutdown
(1000 connection requests)

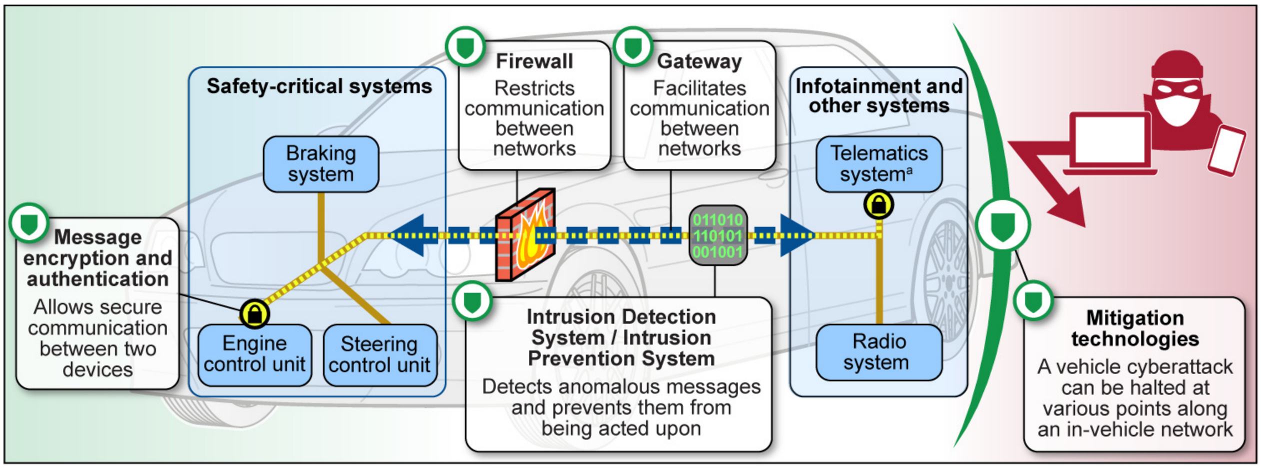
Buffer overflow
(large data packet)

Fake packets
(sender = dron)

https://youtu.be/0Ihin_9wVuA

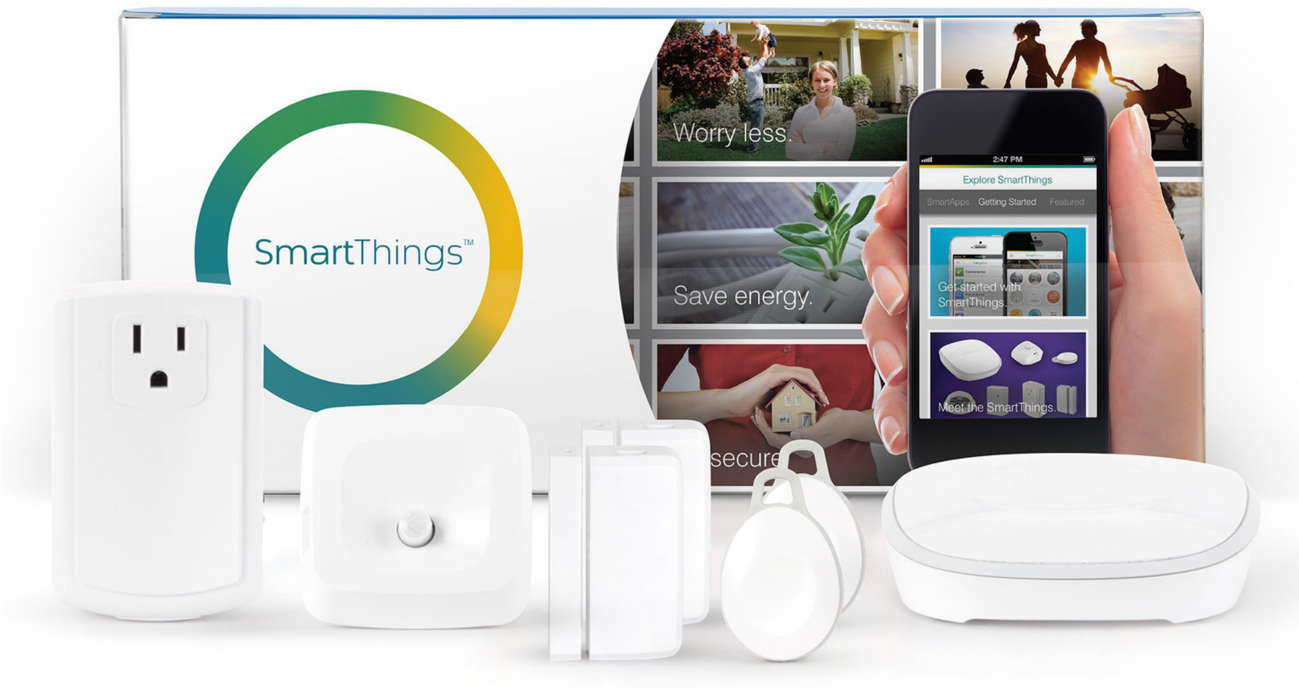
Seguridad

Vehículos: Mitigación de vulnerabilidades



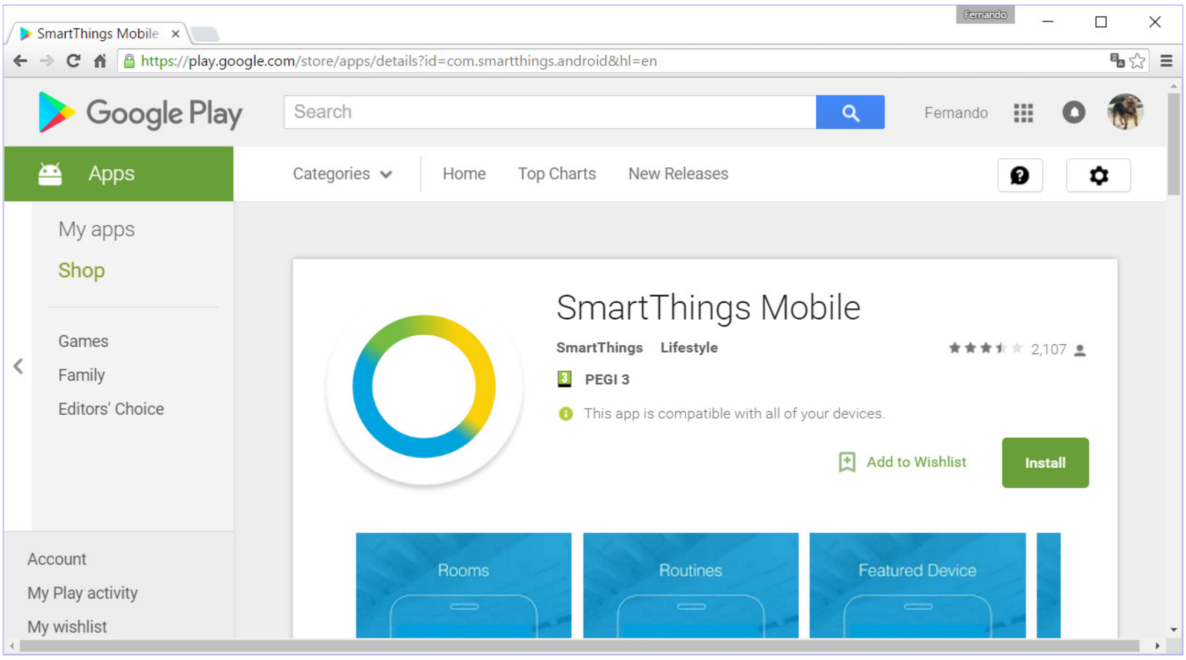
Source: GAO analysis of stakeholder information. | GAO-16-350

Seguridad: Domótica



Seguridad: Domótica

> 100,000 descargas



Seguridad: Domótica

4 successful proof-of-concept attacks (Univ. Michigan)

- A SmartApp that eavesdropped on someone setting a new PIN code for a door lock, and then sent that PIN in a text message to a potential hacker. The SmartApp, which they called a "lock-pick malware app" was disguised as a battery level monitor and only expressed the need for that capability in its code.
- An existing, highly rated SmartApp could be remotely exploited to virtually make a spare door key by programming an additional PIN into the electronic lock. The exploited SmartApp was not originally designed to program PIN codes into locks.

Seguridad: Domótica

4 successful proof-of-concept attacks (Univ. Michigan)

- One SmartApp could turn off "vacation mode" in a separate app that lets you program the timing of lights, blinds, etc., while you're away to help secure the home.
- A fire alarm could be made to go off by any SmartApp injecting false messages.

Seguridad: Domótica

¿Causas?

- “Over-priviledge” (>40% apps): the platform grants its SmartApps too much access to devices and to the messages those devices generate,
  e.g. eavesdrop on setting of lock PIN codes.
- It is possible for app developers to deploy an authentication method called OAuth incorrectly.
  This flaw, in combination with SmartApps being over-privileged, allowed the hackers to program their own PIN code into the lock—to make their own secret spare key.

Seguridad: Domótica

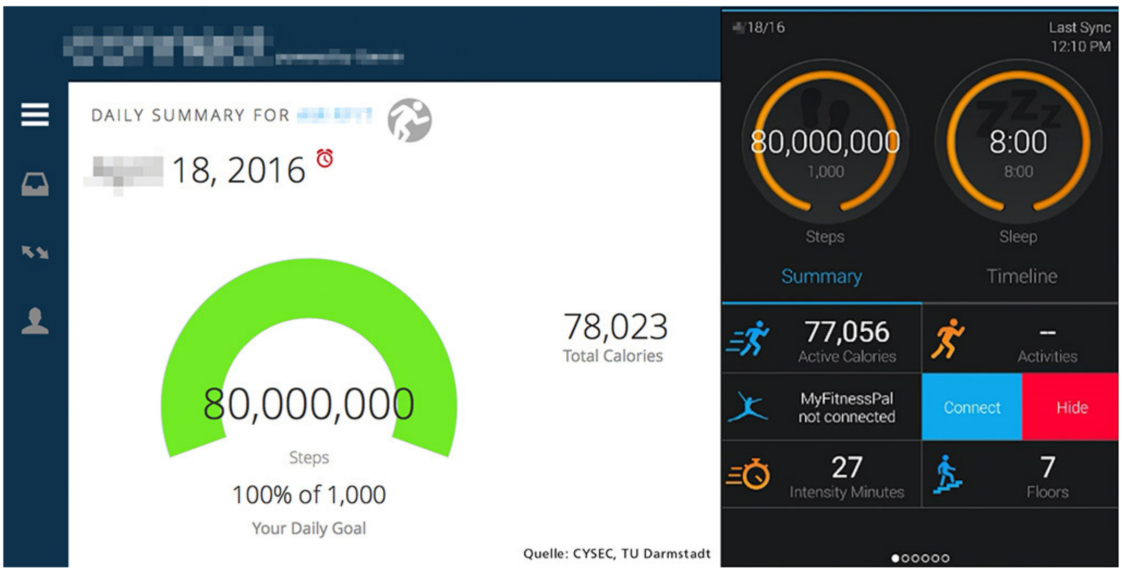
¿Causas?

- The "event subsystem" on the platform is insecure. This is the stream of messages devices generate as they're programmed and carry out those instructions. The researchers were able to inject erroneous events to trick devices. That's how they managed the fire alarm and flipped the switch on vacation mode.

Problemas comunes a distintas plataformas y al IoT…

Seguridad: Fitness trackers

Man-in-the-middle attack



Seguridad: Fitness trackers

Man-in-the-middle attack

	Fitness Tracker	Attack time Year 2016	On Market from Year	Injecting False Data	Uses HTTPS	Encrypts Data	Data Integrity	Proprietary Coding	SSL Pinning	Cloud Services	
										Stores Data	Web Interface
1	Garmin Vivosmart HR	April	2015	✓	✓	✗	✓	✓	✗	✓	✓
2	Garmin Vivofit2	April	2015	✓	✓	✗	✓	✓	✗	✓	✓
3	Garmin Vivofit	April	2014	✓	✓	✗	✓	✓	✗	✓	✓
4	Polar Electro Loop	June	2013	✓	✓	✗	✗	✓	✗	✓	✓
5	ViFit MEDISANA	June	2014	✓	✓	✗	✓	✗	✗	✓	✓
6	Xiaomi MiBand	May	2014	✓	✓	✗	✗	✗	✗	✓	✗
7	Jawbone UP3	July	2015	✓	✓	✗	✗	✗	✗	✓	✗
8	Jawbone Move UP	July	2014	✓	✓	✗	✗	✗	✗	✓	✗
9	Misfit Shine	June	2013	✓	✓	✗	✗	✗	✗	✓	✓
10	Mio Link	July	2014	✓	✓	✗	✓	✗	✗	✓	✗
11	Withings Pulse	July	2013	✓	✓	✗	✗	✗	✗	✓	✓
12	Runtastic Orbit	July	2014	✓	✓	✗	✗	✗	✗	✓	✓
13	Sony Smartband 2	—	2015	—	—	—	—	—	—	✗	✗
14	Razor Nabu X	—	2015	—	—	—	—	—	—	✗	✗
15	Technaxx 39	—	2015	—	—	—	—	—	—	✗	✗
16	Technaxx 37	—	2015	—	—	—	—	—	—	✗	✗
17	Oregon Dynamo 2+	—	2014	—	—	—	—	—	—	✗	✗

— Not Applicable

Quelle: CYSEC, TU Darmstadt

Seguridad

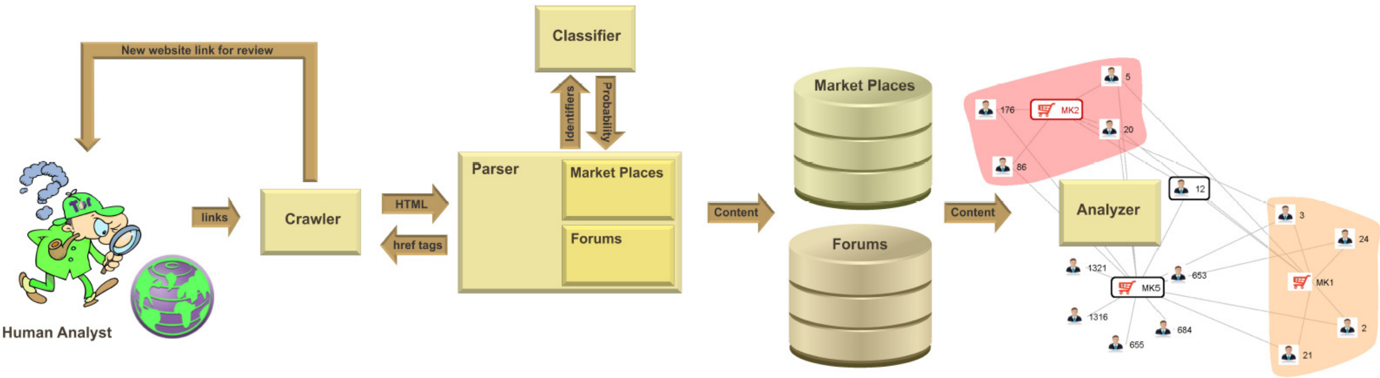
Dispositivos móviles, vehículos autónomos, IoT…

What's needed is the capability to update software routinely in order to patch newly found vulnerabilities, similar to the way Microsoft's Windows Update works.

-- Stefan Savage, UCSD
2015 ACM-Infosys Foundation Award

Seguridad

Machine-Learning Algorithm Combs the Darknet for Zero Day Exploits, and Finds Them



"The first machine-based search of online hacker marketplaces identifies over 300 significant cyberthreats every week." – MIT TR, August 2016

Darknet and Deepnet Mining for Proactive Cybersecurity Threat Intelligence, arXiv, July 2016 arxiv.org/abs/1607.08583

Seguridad

IA (Machine Learning)

e.g Google's Android Security Team
Baidu (malware identification)
Deep Instinct (security startup)
Cylance (security startup)

Google's Training Its AI to Be Android's Security Guard
https://www.wired.com/2016/06/googles-android-security-team-turns-machine-learning/