Universitat Oberta de Catalunya

Universitat Autònoma de Barcelona

UNIVERSITAT ROVIRA i VIRGILI

Universitat de les Illes Balears

Seguridad en Smartphones: Análisis de riesgos, de vulnerabilidades y auditorías de dispositivos

Título: Trabajo Final de Master, Master Interuniversitario en Seguridad de las TIC

Autor: Carlos García Altarejos, carlosg.altarejos@gmail.com, charly84@uoc.edu

Tutor: Marco Antonio Lozano Merino, mlozanome@uoc.edu

Fecha: 12 de enero de 2017

RESUMEN

Con el presente trabajo se pretende desarrollar una metodología para realizar un análisis de riesgos de vulnerabilidades y auditorias de dispositivos, que pueda servir para prevenir estas amenazas y protegerse frente a ellas. El principal objetivo se centra en el estudio de las herramientas y técnicas para detectar las amenazas y las vulnerabilidades que afectan a estos dispositivos.

La metodología está estructurada en cinco fases que son: identificación, análisis, acceso, resultados e informe. Con estas se pretende poder llevar a cabo un análisis para identificar e informar fallas en los dispositivos y en los procesos tecnológicos. Enfocado a la protección total de los recursos que estén expuestos a posibles amenazas de seguridad informática.

Este tipo de análisis no suele incluir, las etapas relacionadas con la explotación de las vulnerabilidades identificadas, sino que solo trabaja sobre la correcta identificación de las mismas. Los sistemas operativos móviles que se estudiarán son Android(Google) e IOS(Apple).

Por último, en las distintas fases de este proyecto se estudian y analizan herramientas de tipo comercial y Open Source, que serán de gran utilidad a la hora de realizar los análisis sobre los correspondientes dispositivos móviles.

ABSTRACT

This project aims to develop a methodology to perform vulnerability risk analysis and device audits, which can be used to prevent and protect against these threats. The main objective is to study the tools and techniques to detect the threats and vulnerabilities that affect these devices.

The methodology is structured in five stages: identification, analysis, access, results and report. These are intended to be able to carry out an analysis to identify and report failures in devices and in technological processes. Focused on the total protection of resources that are exposed to potential security threats.

This type of analysis usually focus on the correct identification of vulnerabilities, but it typically does not include the stages related to the exploitation of said identified vulnerabilities. The mobile operating systems to be studied are Android (Google) and IOS (Apple).

Finally, the various stages of this project study and analyze tools of commercial type and OpenSource, which will be very useful when performing the analysis on the corresponding mobile devices.

INDICE DE CONTENIDOS

PLAN DE TRABAJO

Propósito

La proliferación de los smartphones debido a su avance tecnológico, la facilidad de acceso a la red gracias a la mejora en las infraestructuras y el ancho de banda de sus conexiones, han impactado de manera muy significativa en su uso en todos los ámbitos. Cada vez es más frecuente que una empresa se apoye en soluciones tecnológicas para facilitar y promover el acceso a la información y a los recursos corporativos. Y en el ámbito domestico, utilizamos nuestros smartphones para una gran variedad de actividades como son el acceso a redes sociales, correo electrónico, compras online, transacciones bancarias, navegación web, almacenamiento de documentos, etc.

Por estos motivos, debemos adoptar precauciones para garantizar que nuestros teléfonos e información estén seguros, ya que existen una gran cantidad de amenazas.

Muchas de las amenazas encontradas en la actualidad buscan el acceso a datos confidenciales del usuario, sin que él mismo sea consciente de ello, para el robo de su información y datos personales. Sin embargo hay amenazas que tienen otros objetivos, como la suscripción a servicios de pago, suplantación de identidad, botnets que buscan convertir en zombi al dispositivo, adware para el envío de publicidad no deseada o ransomware que cifra la información y solicita un pago como rescate para recuperar los datos, etc.

Con el presente trabajo se pretende desarrollar una metodología para realizar un análisis de riesgos de vulnerabilidades y auditorias de dispositivos, que pueda servir para prevenir estas amenazas y protegerse frente a ellas.

Objetivos

El principal objetivo de este proyecto es el estudio de las herramientas y técnicas para detectar las amenazas y las vulnerabilidades que afectan a los Smartphones. Los sistemas operativos móviles que se tratarán serán Android(Google) e IOS(Apple).

La lista de objetivos a alcanzar es la siguiente:

1. Entender el concepto de vulnerabilidad de seguridad y su contexto.
2. Conocer como identificar las vulnerabilidades y como se catalogan.
3. Aprender a gestionar las vulnerabilidades.
4. Estudiar las herramientas de análisis de vulnerabilidades más comunes para smartphones.
5. Estudiar los sistemas operativos Android e IOS.
6. Identificar los diferentes tipos de amenazas que afectan a los smartphones.
7. Conocer los mecanismos básicos de prevención contra las distintas amenazas.
8. Obtener una visión completa de las estrategias de detección de amenazas de software.
9. Conocer los mecanismos y herramientas para conseguir Root en Android y jailbreak en IOS.

10. Obtener información de los análisis sobre los dispositivos estudiados.
11. Estudiar las metodologías a seguir para la realización del proceso de análisis y auditoria de smartphones.
12. Desarrollo de una memoria final.

Metodologías y estándares

A nivel mundial existen estándares relacionados con el análisis de vulnerabilidades, auditoría y pruebas de Intrusión que seguiremos para la realización de este proyecto y que enumeramos a continuación:

- BS 7799 e ISO 17799 (Internacional).
- Open Source Security Testing Methodology Manual.
- Federal Information System Controls Audit Manual (FISCAM).
- COBIT(Objetivos de Control de la Tecnológica de la Información).
- ISO 17799(Norma internacional que ofrece recomendaciones para realizar la gestión de la seguridad de la información).
- ISO 27002 (Código de buenas prácticas).
- ISO 27007(Guía para auditar un SGSI).
- ISACA(Asociación de Auditoría y Control de Sistemas de Información).
- ITIL (Biblioteca/Guía de Infraestructura de Tecnologías de la Información) estándar mundial de facto en la Gestión de Servicios Informáticos. Útil para las organizaciones en todos los sectores para consulta, educación y soporte de herramientas de software, de libre utilización.

Estructura del trabajo

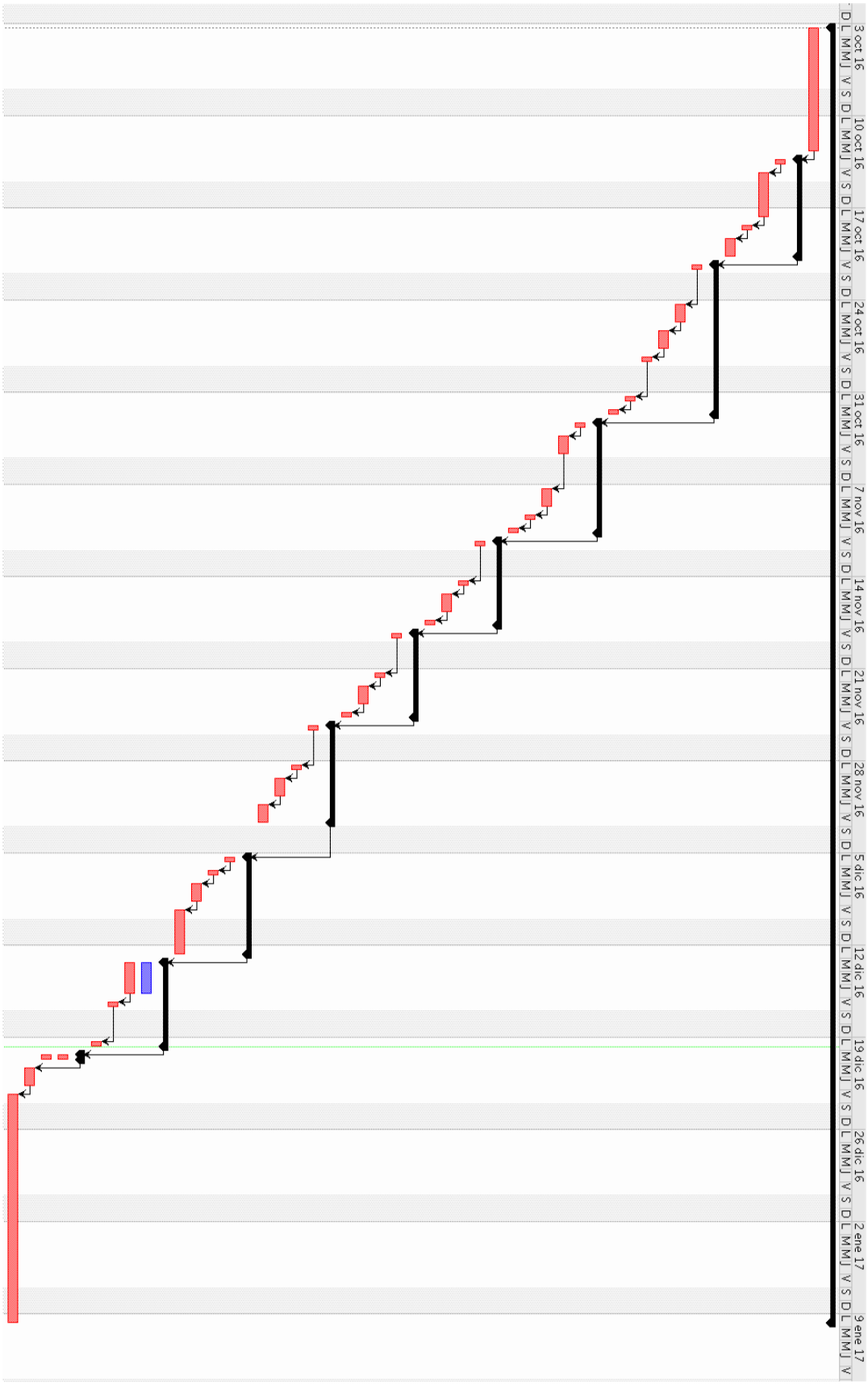
Los diferentes apartados del proyecto son los siguientes:

1. **Desarrollo del plan de trabajo:** En este apartado se identifican los procesos a seguir para el desarrollo del proyecto de forma exitosa.
2. **Metodología para el análisis de vulnerabilidades:** Se pretende desarrollar una metodología, identificando y definiendo las diferentes fases del proceso para su correcta ejecución.
3. **Análisis de dispositivos móviles IOS:** En este apartado se estudiará el sistema operativo IOS, su estructura interna y su funcionamiento.
4. **Análisis de dispositivos móviles Android:** De igual forma que en apartado anterior, se estudiará el sistema operativo Android, como funciona y su estructura interna.
5. **Escaneo de dispositivos móviles IOS:** Se centra en el descubrimiento y la exploración del dispositivo. Es decir, se analizarán que servicios tiene en escucha, así como las diferentes herramientas para su análisis.

6. Escaneo de dispositivos móviles Android: De igual forma que antes se analizarán que servicios tiene en escucha, así como las diferentes herramientas para su análisis.
7. Evaluación y test de intrusión en dispositivos móviles IOS: El “Pentesting” o test de intrusión es un método de auditoría mediante el cual se intenta acceder a los sistemas, para comprobar el nivel de resistencia a la intrusión no deseada. Llegado a este punto se estudiarán los diferentes métodos de acceso al dispositivo IOS, las herramientas para su estudio, detección de vulnerabilidades y su posterior explotación.
8. Evaluación y test de intrusión en dispositivos móviles Android: En este apartado se estudiarán los diferentes métodos de acceso al dispositivo Android, las herramientas para su estudio y su posterior explotación.
9. Informe de auditoria: En este informe se pretende mostrar las diferentes vulnerabilidades de 2 dispositivos móviles, IOS y Android, las amenazas y riesgos a las que se exponen y como protegerse de ellas.
10. Memoria final.

Planificación temporal

	Nombre	Duracion	Inicio	Terminado
1	**Inico del TFM**	71 days	**3/10/16 8:00**	**9/01/17 17:00**
2	Plan de trabajo	8 days	3/10/16 8:00	12/10/16 17:00
3	**Metodología para el análisis de vulnerabilidades**	6 days	**13/10/16 8:00**	**20/10/16 17:00**
4	Definición de análisis y auditoría de smartphones	1 day	13/10/16 8:00	13/10/16 17:00
5	Fases de la metodología	2 days	14/10/16 8:00	17/10/16 17:00
6	Definición y análisis de amenazas de seguridad	1 day	18/10/16 8:00	18/10/16 17:00
7	Descripción del laboratorio	2 days	19/10/16 8:00	20/10/16 17:00
8	**Análisis de dispositivos móviles IOS**	8 days	**21/10/16 8:00**	**1/11/16 17:00**
9	Descripción del sistema operativo iOS	1 day	21/10/16 8:00	21/10/16 17:00
10	Estructura del sistema operativo iOS	2 days	24/10/16 8:00	25/10/16 17:00
11	Seguridad en iOS	2 days	26/10/16 8:00	27/10/16 17:00
12	Arquitectura de seguridad en iOS	1 day	28/10/16 8:00	28/10/16 17:00
13	Seguridad de la red en iOS	1 day	31/10/16 8:00	31/10/16 17:00
14	Servicios de Internet	1 day	1/11/16 8:00	1/11/16 17:00
15	**Análisis de dispositivos móviles Android**	7 days	**2/11/16 8:00**	**10/11/16 17:00**
16	Descripción del sistema operativo Android	1 day	2/11/16 8:00	2/11/16 17:00
17	Estructura del sistema operativo Android	2 days	3/11/16 8:00	4/11/16 17:00
18	Seguridad en Android	2 days	7/11/16 8:00	8/11/16 17:00
19	Arquitectura de seguridad en Android	1 day	9/11/16 8:00	9/11/16 17:00
20	Seguridad en las aplicaciones Android	1 day	10/11/16 8:00	10/11/16 17:00
21	**Escaneo de dispositivos móviles IOS**	5 days	**11/11/16 8:00**	**17/11/16 17:00**
22	Estudio de herramientas escaneo	1 day	11/11/16 8:00	11/11/16 17:00
23	Exploración del dispositivo	1 day	14/11/16 8:00	14/11/16 17:00
24	Análisis de la información obtenida	2 days	15/11/16 8:00	16/11/16 17:00
25	Elaboración del informe	1 day	17/11/16 8:00	17/11/16 17:00
26	**Escaneo de dispositivos móviles Android**	5 days	**18/11/16 8:00**	**24/11/16 17:00**
27	Estudio de herramientas escaneo	1 day	18/11/16 8:00	18/11/16 17:00
28	Exploración del dispositivo	1 day	21/11/16 8:00	21/11/16 17:00
29	Análisis de la información obtenida	2 days	22/11/16 8:00	23/11/16 17:00
30	Informe de resultados	1 day	24/11/16 8:00	24/11/16 17:00
31	**Evaluación y acceso al sistema en IOS**	6 days	**25/11/16 8:00**	**2/12/16 17:00**
32	Estudio de las herramientas a utilizar	1 day	25/11/16 8:00	25/11/16 17:00
33	Identificación de vulnerabilidades y proceso de Jailbreak	1 day	28/11/16 8:00	28/11/16 17:00
34	Métodos de explotación y acceso	2 days	29/11/16 8:00	30/11/16 17:00
35	Análisis de datos	2 days	1/12/16 8:00	2/12/16 17:00
36	**Evaluación y acceso al sistema en Android OS**	6 days	**5/12/16 8:00**	**12/12/16 17:00**
37	Estudio herramientas a utilizar	1 day	5/12/16 8:00	5/12/16 17:00
38	Identificación de vulnerabilidades y Android Rooting	1 day	6/12/16 8:00	6/12/16 17:00
39	Métodos de explotación y acceso	2 days	7/12/16 8:00	8/12/16 17:00
40	Análisis de datos	2 days	9/12/16 8:00	12/12/16 17:00
41	**Informe final y conclusiones**	5 days	**13/12/16 8:00**	**19/12/16 17:00**
42	Dispositivos móviles con Android OS	3 days	13/12/16 8:00	15/12/16 17:00
43	Dispositivos móviles con iOS	3 days	13/12/16 8:00	15/12/16 17:00
44	Conclusiones	1 day	16/12/16 8:00	16/12/16 17:00
45	Trabajo futuro	1 day	19/12/16 8:00	19/12/16 17:00
46	**Anexo**	1 day	**20/12/16 8:00**	**20/12/16 17:00**
47	Medidas de prevención y recomendaciones	1 day	20/12/16 8:00	20/12/16 17:00
48	Plan de respuesta y mitigación en caso de infección	1 day	20/12/16 8:00	20/12/16 17:00
49	Memoria final	2 days	21/12/16 8:00	22/12/16 17:00
50	Presentación, vídeo y defensa	12 days	23/12/16 8:00	9/01/17 17:00


3 oct 16
10 oct 16
17 oct 16
24 oct 16
31 oct 16
7 nov 16
14 nov 16
21 nov 16
28 nov 16
5 dic 16
12 dic 16
19 dic 16
26 dic 16
2 ene 17
9 ene 17

METODOLOGÍA PARA EL ANÁLISIS DE VULNERABILIDADES

Definición de análisis y auditoría de Smartphones

Una auditoría de seguridad informática sobre smartphones trata sobre la evaluación de los dispositivos móviles cuyo fin es detectar errores y fallos en el sistema, y que mediante un informe detallado se muestren los resultados verificados y las medidas y recomendaciones a tomar.

Las auditorías de seguridad permiten conocer cuál es la situación exacta del sistema en cuanto a protección, control y medidas de seguridad.

Los servicios de auditoría pueden ser de distintos tipos, y en nuestro estudio incluiremos el análisis de vulnerabilidades y la descripción del test de intrusión o pruebas de penetración. A continuación los describiremos brevemente.

La metodología de **análisis de vulnerabilidades** informáticas, está enfocada a la protección total de los recursos (en este caso sobre dispositivos móviles) que estén expuestos a posibles amenazas de seguridad informática.

Es un tipo de análisis que busca identificar e informar fallas en los dispositivos y en los procesos tecnológicos. Se trata de **análisis y pruebas** relacionadas con la identificación de puertos abiertos, servicios disponibles y vulnerabilidades conocidas en los sistemas. Todos estos procesos incluyen verificaciones manuales y automáticas, lo que esto permite identificar puntos débiles y realizar un análisis profesional individualizado.

Este tipo de análisis no suele incluir, las etapas relacionadas con la explotación de las vulnerabilidades identificadas, sino que solo trabaja sobre la correcta identificación de las mismas.

Un **test de penetración** se caracteriza por tener un objetivo definido que finaliza cuando el mismo es alcanzado o el tiempo pautado para el desarrollo del análisis se agota. Es un tipo de análisis de seguridad que no solo trata de identificar e informar las debilidades, sino que también intenta explotarlas a fin de verificar fehacientemente los niveles de intrusión a los que se expone el sistema de información analizado.

En este tipo de test se evalúa la seguridad de los sistemas para comprometer los accesos a los diferentes servicios, aplicaciones, datos personales, cuentas de usuario etc., simulando los ataques a los que podrían estar expuestos. Esta actividad, involucra un análisis activo sobre los sistemas, explotando las vulnerabilidades que se puedan encontrar en los mismos.

Las **pruebas de penetración** sirven para:

- Identificar las vulnerabilidades de alto riesgo de un sistema.
- Identificación de las vulnerabilidades que pueden ser difíciles o imposibles de detectar
- Evaluación de la magnitud de las vulnerabilidades encontradas y los impactos operacionales de ataques con éxito.
- Pruebas a la capacidad de las herramientas para detectar las pruebas y responder a los ataques.

- Proporcionar evidencias para lograr la mejora de los dispositivos, herramientas, etc. En lo que se refiere a la seguridad y tecnología.

Asimismo la metodología consiste en los procesos iterativos, debido a que la tecnología nunca deja de evolucionar y cada vez más se generan nuevos riesgos para los dispositivos móviles. La metodología de análisis de vulnerabilidades y auditoría sobre smartphones ha sido estructurada en diferentes módulos.

Fases de la metodología

Llegado a este punto del desarrollo, hemos procedido a diseñar la metodología basándonos en los diferentes modelos existentes. Por tanto hemos conseguido desarrollar esta estructura modificando e incorporando nuevos elementos y valores que nos proporcionan una metodología nueva y consistente, y que a su vez nos permita aplicarla a la realidad, en cuanto a riesgos informáticos en entornos compuestos por smartphones.

El procedimiento se ha llevado a cabo a través de unificar etapas de la metodología básica con similitudes y características propias de cada una de ellas, logrando obtener como resultado cinco fases y con procedimientos implementados en cada uno de ellos. La estructura de la metodología desarrollada en este trabajo la podemos observar en el esquema siguiente:

Fases de la metodología

El esquema con las diferentes fases es el siguiente:

Fase 1: Identificación

- Exploración física del dispositivo.
- Identificación del dispositivo.
- Definición del sistema a auditar.

Fase 2: Análisis

- Escaneo de puertos.
- Detección de servicios.
- Detección del sistema operativo.
- Identificación de servicios inalámbricos.

Fase 3: Acceso

- Vectores de ataque.

- Análisis de actualizaciones.
- Identificación del vulnerabilidades.
- Técnicas de explotación de vulnerabilidades.
- Control y acceso a la información.

Fase 4: Resultados

- Evidencias de accesos alcanzados.
- Auditoría del sistema.

Fase 5: Informes

- Elaboración de informes.
- Medidas y recomendaciones de control.

Fase 1: Identificación

Esta fase consiste en la identificación y definición de los sistemas a auditar. El objetivo es la obtención de información sobre el dispositivo móvil, con el fin de adquirir la máxima información acerca del tipo de dispositivo, del modelo, incluso de la versión del sistema operativo(de manera aproximada).

Para llevar a cabo esta primera fase lo primero que debemos hacer es observar el dispositivo físicamente para poder diferenciar la marca y el modelo. Dependiendo del resultado obtenido podremos continuar con la recolección de información. Por ejemplo, si detectáramos un modelo obsoleto de Iphone, simplemente haciendo un reconocimiento visual del aparato, podríamos saber cual es la última versión de iOS para este dispositivo haciendo una consulta en la propia web de Apple Inc.

Una vez identificado el fabricante, o al menos descartado o confirmado que es un producto de la compañía Apple Inc., si observamos la pantalla inicial del teléfono en su estado bloqueado, podemos llegar a deducir que versión aproximada de sistema operativo tiene instalado el dispositivo.

Fase 2: Análisis

Para la realización de esta fase lo primero que haremos es un análisis de la red. Para esto debemos conectar el dispositivo a una red inalámbrica, donde también tengamos conectado el equipo preparado para el análisis. El análisis de la red consiste en la recolección de datos y la obtención de información acerca de los componentes hardware y software, así como la versión de estos elementos. En el caso del software estaríamos refiriéndonos, por ejemplo, a los servicios de red en escucha o puertos activos. Analizando esta información con las herramientas adecuadas podemos obtener el sistema operativo del smartphone analizado. Incluso si el dispositivo está desbloqueado mediante la técnica de Jailbreak o Rooteo.

El siguiente paso de esta fase de análisis sería la detección de sistemas de comunicación inalámbricos activos en el dispositivo móvil. Dependiendo del modelo de dispositivo, su antigüedad y la configuración que

disponga, podremos obtener más información sobre otros de los servicios de comunicación activos en el smartphone que analizamos. Entre estos podemos identificar a los más comúnmente conocidos como Bluetooth o NFC.

A través de la conexión, mediante el cable de datos y con el software adecuado, con nuestro sistema de análisis podremos obtener información detallada del dispositivo, como el IMEI, número de serie, nombre del dispositivo o versión del SO.

Fase 3: Acceso

Durante esta fase de procederá a identificar las vulnerabilidades del dispositivo y a su posterior explotación, siempre en un entorno de pruebas controlado. La detección de vulnerabilidades se realiza tanto de forma automática como de forma manual y, en ambos casos, se lleva a cabo una fase de validación de las vulnerabilidades identificadas para descartar falsos positivos.

En esta fase se analizarán las actualizaciones pendientes que tiene el sistema operativo, y con esta información podremos averiguar que vulnerabilidades le afectan y la gravedad de estas.

Una vez detectadas las vulnerabilidades y la gravedad de estas, buscaremos un vector de ataque y los exploits específicos para explotarlas o herramientas especializadas para ello y poder así, acceder al terminal y obtener la información privada que nos hayamos marcado como objetivo, o en todo caso obtener el acceso al dispositivo para continuar con la intrusión.

Fase 4: Resultados

Una vez alcanzada esta fase, con la información y evidencias obtenidas en las fases anteriores procederemos a enumerar y documentar los métodos que se pueden utilizar y los resultados que podemos obtener.

En esta fase describiremos de manera simplificada los procesos realizados hasta llegar a obtener el objetivo, así como las técnicas utilizadas y las herramientas asociadas a estas, en caso que hubieran sido necesarias.

Fase 5: Informe

Esta es la última fase de la metodología, donde una vez obtenidos los resultados y verificados, se emite un informe indicando el establecimiento de las medidas preventivas de refuerzo y/o corrección a los problemas obtenidos, junto a las posibles soluciones a estos, y a las recomendaciones de seguridad y buenas prácticas, que permita a los usuarios mejorar la seguridad de sus dispositivos aprendiendo de los errores cometidos con anterioridad.

Definición y análisis de amenazas de seguridad

La tecnología, a través de los smartphones, está cada vez más presente en casi todos los aspectos de nuestra vida personal y profesional, mejorándola y generando cada vez más oportunidades para la innovación, aunque también dejando más espacio a las amenazas.

A medida que los dispositivos móviles se conviertan en una prolongación del individuo, contribuyendo a que nuestro entorno sea más inteligente, más consciente del contexto y más conectado, deberemos esperar cambios a todos los niveles.

El valor de la información almacenada y en tránsito aumenta con rapidez, lo que favorece el nacimiento de nuevos mercados, crea una necesidad de dispositivos conectados con seguridad, transfiere datos fiables a la nube y aporta valor gracias a herramientas analíticas. Como cualquier otro activo de valor, la información también atrae la atención de los ciberdelincuentes, que buscan nuevas formas de apoderarse de ella y aprovecharla en su beneficio. A menudo pensamos que los ataques informáticos son obra del crimen organizado y de otros tipos de ciberdelincuentes, pero también pueden estar dirigidos por hacktivistas, gobiernos de países y otros individuos cuyo objetivo no es necesariamente obtener una rentabilidad financiera directa. Asimismo, asistimos a la aparición de ciberataques personalizados y privados, que pueden ser obra de cualquier persona con intención de causar daño.

Una amenaza es la posibilidad de ocurrencia de cualquier tipo de evento o acción que puede producir un daño (material o inmaterial) sobre los elementos de un sistema, y en el caso de los Smartphones, sobre los elementos relacionados con la información y el propio dispositivo hardware.

Las amenazas y los consecuentes daños que puede causar una acción exitosa de este tipo, pueden afectar a la confidencialidad, integridad, disponibilidad y autenticidad de los datos que almacenamos en nuestros terminales móviles.

Existen amenazas de diferentes tipos que generalmente se agrupan en:

1. Criminales: Son acciones causadas por la intervención humana con una intencionalidad delictiva, que suelen tener diferentes objetivos. Estas son el principal grupo que incluye tanto el robo, fraude, espionaje, virus, etc.
2. Físicas: Son las amenazas causadas por las acciones directas sobre los dispositivos, causadas por la intervención humana, como podría ser un accidente físico que afectara al terminal, una inundación o una sobrecarga eléctrica.
3. Negligentes: Son las acciones, decisiones u omisiones por parte de los usuarios del dispositivo. Son menos predecibles ya que están relacionadas con el comportamiento humano. A este grupo pertenecen las amenazas como la pérdida de información debido a un mal uso por falta de capacitación o mal manejo del dispositivo.

Clasificación de las amenazas en Smartphones

A continuación vamos a identificar de manera más detallada las amenazas a las que están expuestos los smartphones:

- Pérdida de información: Debido al uso generalizado de los smartphones y su gran variedad de usos, en ellos almacenamos información que, por múltiples razones, podemos llegar a perder. Ya sea por un fallo hardware, un error humano, un fallo software, etc.
- Robo del dispositivo: Cualquier usuario que utilice su smartphone en lugares públicos, está expuesto a este tipo de amenaza.
- Rotura del dispositivo: Este tipo de amenaza puede ser debido a diferentes motivos. Por ejemplo un fallo hardware, como el reciente problema de los Samsung Galaxy Note 7 que explotan las baterías, una mala caída del dispositivo contra el suelo o sumergirlo en algún líquido. Con lo cual, podrían dejar nuestro terminal inservible.
- Robo de credenciales: Es una de las principales amenazas. El objetivo es obtener las cuentas de usuario y contraseñas de las aplicaciones instaladas e incluso del propio smartphone. Pueden llevarse a cabo mediante ataques de phising e ingeniería social, infección de malware o técnicas de MitM.
- Suplantación de identidad o Spoofing: Este tipo de amenaza consta de diferentes subtipos, y se utiliza para engañar a la víctima con diferentes objetivos. Por ejemplo el Web Spoofing trata de suplantar una página web real, redirigiendo la conexión de una víctima a través de una página falsa hacia otras páginas WEB con el objetivo de obtener información de dicha víctima. O el email spoofing que es la suplantación de la dirección de correo electrónico de otras personas o empresas, que tiene el mismo objetivo que el anterior.
- Acceso a datos confidenciales como conversaciones, imágenes, vídeos o documentos: En nuestros dispositivos móviles inteligentes almacenamos todo tipo de información personal y confidencial que, en caso de sufrir un ataque podría verse comprometida.
- Robo de información: Este tipo de amenaza junto con el robo de credenciales es una de las más importantes. Ya sea de tipo personal o profesional, pero la información que almacenamos en nuestros smartphones está expuesta a ser sustraída, utilizando diferentes técnicas de hacking.
- Control remoto del dispositivo sin autorización: Existen diferentes tipos de malware que tienen como objetivo la obtención del control remoto de los smartphones. Una vez se hacen con el control pueden llevar a cabo diferentes acciones libremente. Normalmente son utilizados junto a otros terminales para formar una red de bots o botnet. Y son utilizadas para realizar ataques DDoS, envío de Spam, robo y minería de bitcoins o cualquier otro fin que pueda aportar beneficios económicos al creador.
- Deterioro del terminal: Es uno de los motivos que nos hacen cambiar de terminal, por ejemplo por problemas de autonomía de la batería o por problemas del hardware debido a su uso y desgaste. Al fin y al cabo nuestro dispositivo puede verse deteriorado por diferentes motivos.

- Incremento de la factura: Es uno de los objetivos de algunos de los ataques que reciben los smartphones. Mediante la infección de un dispositivo y su posterior control, mediante la realización de llamadas a números de pago que hacen que nuestra factura se vea afectada.
- Descarga de aplicaciones no deseadas: Esta amenaza suele tener como objetivo el Spam publicitario, es decir publicitar Apps, aumentando el número de descargas en el mercado. Aunque también es utilizada en botnets para otros fines maliciosos.
- Infección por virus o malware: Mediante diferentes vías de acceso nuestro dispositivo puede ser infectado por este tipo de software malicioso. Existen varios tipos de malware, los más propagados en los últimos años son los troyanos, spyware, adware, rootkits, gusanos, ransomware, etc. Una vez infectado los riesgos a los que se ve afectado un smartphone son muy amplios, dependiendo del objetivo del código malicioso.
- Ataques de phising: El phishing es una técnica que consiste en engañar al usuario para robarle información confidencial, claves de acceso, etc, haciéndole creer que está en un sitio de total confianza. Lo más utilizado hasta hace poco han sido los correos electrónicos para lanzar este tipo de ataques, pero con el uso masivo de redes sociales y smartphones con conexión a Internet, las vías de ataque se están multiplicando.
- Sideloading mediante certificados iOS falsos: Esta amenaza para tu smartphone parte de los certificados de distribución para hacer 'sideloading' de una aplicación, es decir, dejando a un lado el proceso de validación oficial de la tienda de aplicaciones de Apple, permitiendo una descarga directa en el dispositivo.
- Perfiles iOS maliciosos: Estos perfiles falsos eluden los mecanismos de seguridad típicos, permitiendo, que el atacante modifique el recorrido del tráfico, de tal forma que este sea redirigido desde el dispositivo móvil a un servidor controlado por él.
- Utilización de sistemas no seguros: Por desgracia, no siempre los sistemas operativos de smartphones y las aplicaciones que utilizamos son seguros, bien porque no pueden estar actualizados a la última versión, por el uso de protocolos no seguros o por errores en el desarrollo. Esto conlleva el riesgo, por ejemplo, de utilizar protocolos de comunicación no seguros o versiones de SO que tienen agujeros de seguridad graves.
- Fragmentación del SO Android: Android, el sistema operativo de Google y el más extendido del mundo. Lo cierto es que, tiene sus puntos negativos y uno de ellos es la fragmentación. La fragmentación se debe a que desde el lanzamiento de Android hasta la fecha han aparecido un gran número de fabricantes apostando por él. Por lo que actualmente el desarrollo y el soporte de Android, tanto por Google como por los fabricantes tiene un margen de dos años de actualizaciones, y esto con el paso del tiempo se agrava y es prácticamente imposible que estos no terminen en obsolescencia.
- Vigilancia: Los Smartphones tienen acceso a Internet y están equipados con GPS y cámara de fotos y video. Una aplicación instalada en nuestro dispositivo podría tener acceso a estos datos, y si esta ha sido diseñada con fines maliciosos, podríamos estar vigilándonos remotamente sin que lo supiéramos. Accediendo a nuestra ubicación, audio, mensajes, cámara o registros de llamadas entre otras cosas.

Para entender los tipos de amenazas actuales, es necesario conocer el significado de malware, los tipos de malware existentes y cómo funcionan.

"El malware (del inglés "malicious software"), también llamado badware, código maligno, software malicioso, software dañino o software malintencionado, es un tipo de software que tiene como objetivo infiltrarse o dañar una computadora o sistema de información sin el consentimiento de su propietario." Fuente: Wikipedia.org

El software se considera malware en función de los efectos que provoque en un sistema. El término malware incluye:

1. Virus: Un virus informático es un programa o software que se autoejecuta y se propaga insertando copias de sí mismo en otro programa o documento. El objetivo principal de un virus es infectar.
2. Ransomware: Es un tipo de malware que ha proliferado mucho en los últimos años. El ransomware es un tipo de malware que bloquea un dispositivo o cifra datos en él y luego pide un rescate para desbloquearlo o descifrar los datos. El ransomware se difunde normalmente usando tácticas de ingeniería social.
3. Gusanos de red: Este tipo de malware usa los recursos de red para distribuirse. Su nombre implica que pueden penetrar de un equipo a otro como un gusano. Lo hacen por medio de correo electrónico, sistemas de mensajes instantáneos, redes de archivos compartidos (P2P), canales IRC, redes locales, redes globales, etc. Su velocidad de propagación es muy alta.
4. Caballos de Troya o troyanos: Esta clase de programas maliciosos incluye una gran variedad de programas que efectúan acciones sin que el usuario se dé cuenta y sin su consentimiento. Recolectan datos y los envían a los criminales, destruyen o alteran datos con intenciones delictivas, causando desperfectos en el funcionamiento del dispositivo o usan los recursos del sistema para fines criminales, como hacer envíos masivos de correo no solicitado. Suelen estar camuflados y se propagan bajo la vela de algún software "deseable".
5. Spyware: Software que permite colectar la información sobre un usuario/organización de forma no autorizada. Su presencia puede ser completamente invisible para el usuario. Pueden colectar los datos sobre las acciones del usuario, el contenido del dispositivo, software instalado, registro de llamadas, mensajes, etc.
6. Adware: Muestran publicidad al usuario. La mayoría de programas adware son instalados por software distribuido gratuitamente. La publicidad aparece en la interfaz. A veces pueden recoger y enviar los datos personales del usuario.
7. Riskware: No son programas maliciosos pero contienen una amenaza potencial. En ciertas situaciones ponen sus datos a peligro. Incluyen programas de administración remota, marcadores, etc.
8. Rootkits: Un rootkit es una colección de programas usados por un atacante para evitar ser detectado mientras busca obtener acceso no autorizado a un sistema. Esto se logra de dos formas: reemplazando archivos o bibliotecas del sistema, o instalando un módulo de kernel. El atacante instala el rootkit después, obteniendo un acceso similar al del usuario: por lo general, crackeando una contraseña o explotando una vulnerabilidad, lo que permite usar otras credenciales hasta conseguir el acceso de administrador.

Para que un atacante pueda acceder a un smartphone y comprometer su seguridad, lo más común es infectar el dispositivo mediante algún tipo de malware utilizando una vía de contagio o interceptar las comunicaciones. A continuación definiremos las vías de acceso y contagio más comunes en los smartphones y las medidas de seguridad y prevención de riesgos de estas. Actualmente las más comunes son a través de:

- Vulnerabilidades: Son fallos de seguridad en el software del dispositivo. Pueden ser provocadas por un desarrollo inadecuado del software. Estas vulnerabilidades pueden estar presentes en los sistemas operativos y en las aplicaciones que se instalan. Para solucionarlas, hay que instalar las actualizaciones de seguridad que los desarrolladores de software publican cuando los fallos son detectados y corregidos, tanto para el sistema operativo como para las aplicaciones instaladas. Aún así podríamos estar expuestos a vulnerabilidades que todavía no han sido publicadas, lo que se conoce como 0-day. Un 0-day es una nueva vulnerabilidad para la cual no se crearon parches o revisiones, y que se emplea para llevar a cabo un ataque. En resumen, un 0-day se refiere a un problema que aún no cuenta con una solución. Esta es una de las razones por las cuales recomendamos tener una protección en varias capas.
- Ficheros: Recibidos a través de Bluetooth, correo electrónico, mensajería instantánea, redes sociales o descargados de Internet. Para prevenir esta vía de contagio, solo hay que descargar o aceptar ficheros de sitios de total confianza o de otros dispositivos conocidos. Estos ficheros pueden ser de varios tipos.
- Navegación web: Se han detectado algunos casos de infección al navegar en alguna página web manipulada para que detecte nuestro sistema, busque alguna vulnerabilidad y nos infecte con el malware. Para mitigar esta amenaza, se deben usar navegadores o programas que analicen las páginas web antes de cargarlas. Además, se debe navegar solo por sitios de confianza y ser muy cautos con los enlaces recibidos vía SMS/MMS, email, redes sociales y mensajería instantánea, ya que podrían llevarnos a una página web fraudulenta.
- Aplicaciones fraudulentas: Hay aplicaciones que realmente son malware camuflado. Este tipo de aplicaciones, normalmente, están en mercados o tiendas no oficiales aunque, en algunos casos, se han encontrado aplicaciones maliciosas en los canales oficiales de distribución. La forma de prevenir esta infección es descargar aplicaciones solo de los sitios oficiales, y que además tengan una valoración positiva por los usuarios ya que, de lo contrario, solo tendremos la confianza que nos ofrezca el desarrollador de la aplicación.
- Redes WI-FI no protegidas: Existen puntos Wi-Fi gratuitos que son utilizados para espiar las conexiones de la red mediante ataques Man-in-the-Middle (MitM). Por lo tanto no hay que usar estos servicios Wi-Fi para acceder a sitios web en los que se requiera introducir información personal o confidencial. También hay que asegurarse de que en caso de usar este tipo de conexiones, las páginas a las que accedemos dispongan de cifrado en las comunicaciones (HTTPS).
- Redes 2G/3G/4G: A través de estos estándares de comunicación también pueden llevarse a cabo ciertos ataques. En redes 2G la seguridad es muy baja y sus protocolos y algoritmos de cifrado presentan multitud de vulnerabilidades permitiendo a un atacante realizar escuchas pasivas y escuchas activas pudiendo llevar a cabo la suplantación de identidad, obtención del contenido de SMS, y ataques de denegación de

servicio. En el caso del 3G y 4G el nivel de seguridad es mayor, pero todavía se podría llevar a cabo ataques de denegación de servicio, ataques MitM, incluso realizar escuchas utilizando técnicas de Eavesdropping.

- Ataques a NFC: La tecnología NFC, Near Field Communication, o lo que es lo mismo, comunicación de campo cercano, es un sistema de comunicación inalámbrico de corto alcance que nos permite el envío de datos entre dispositivos a través de un chip que incluyen nuestros smartphones. No todos los dispositivos cuentan con ella. Se usa mayoritariamente para realizar pagos sin necesidad de usar tarjetas o dinero en efectivo, intercambio de datos y automatización de tareas. Las principales amenazas detectadas hasta el momento son el robo de información mediante eavesdropping, corrupción o modificación de los datos en tránsito y los ataques MitM.
- Acceso remoto debido al "Jailbreak" o el "rooteado": Otro factor a tener en cuenta en la seguridad de este tipo de dispositivos, es la eliminación de ciertas protecciones que hay en los sistemas operativos mediante Jailbreak o rooteado. Mediante el primero se consigue que el sistema operativo IOS (Apple) pueda instalar aplicaciones que no están firmadas, y por tanto, que no están en la tienda de Apple (Applestore). Esto supone una reducción en la seguridad del dispositivo debido a que solamente las aplicaciones de la tienda oficial se han verificado y aportan un nivel de confianza alto. En el caso de los dispositivos Android, el sistema puede instalar aplicaciones firmadas (configuración por defecto) o se puede configurar para poder instalar cualquier aplicación, aunque no sea de confianza. El proceso de "rooteado" consiste en poder darle a una aplicación permisos "especiales", lo que implica mayor riesgo en la operativa que pueda llevar a cabo esta aplicación. Estas acciones incrementan la posibilidad de instalación de un software malicioso en un equipo, ya que disminuyen el nivel de seguridad del sistema. Por ese motivo se desaconsejan totalmente.

Descripción del laboratorio

Realizar auditorias y análisis de seguridad en smartphones es una tarea compleja, involucra un proceso en donde se realizan distintos tipos de tareas que identifican, en una infraestructura objetivo, las vulnerabilidades que podrían explotarse y los daños que podría causar un atacante. En otras palabras, se realiza un proceso de auditoria(hacking ético) para identificar qué incidentes podrían ocurrir antes de que sucedan y, posteriormente, reparar o mejorar el sistema, de tal forma que se eviten estos ataques.

Para poder realizar el estudio de forma profesional, es necesario sumar a los conocimientos de hacking ético otros aspectos fundamentales como la programación, metodologías, documentación y la utilización de herramientas, que formarán parte del proceso de auditoria.

A continuación nombraremos el equipamiento necesario y algunas de las herramientas que utilizaremos y una breve descripción sobre ellas:

Equipo técnico:

- Ordenador(PC, Mac o Linux).
- Conexión a Internet.

- Cables de datos de enlace para dispositivos IOS y Android.

Herramientas que vamos a necesitar para la realización de las pruebas pertinentes, correspondientes a cada fase de la metodología definida:

- Nmap: Es una herramienta gratuita de escaneo de redes que permite identificar qué servicios se están ejecutando en un dispositivo remoto, así como la identificación de estos, y existencia de filtros o firewalls, entre otros.
- Nessus: Herramienta que sirve para identificar vulnerabilidades en los servicios. Nessus posee una extensa base de datos de vulnerabilidades conocidas en distintos servicios y, por cada una de éstas, posee plugins que se ejecutan para identificar si la vulnerabilidad existe o no en determinado equipo objetivo. Al ejecutarse Nessus se probarán miles de vulnerabilidades y se obtendrá como resultado un listado de las vulnerabilidades que fueron identificadas.
- Tools para rootear dispositivos Android: Frama Root y Kingo Root son herramientas que permiten rootear un dispositivo Android, es decir obtener permisos de root para poder acceder a todo el contenido del dispositivo.
- Tools para realizar Jailbreak de dispositivos IOS: El jailbreak en IOS es equivalente al root de Android. El jailbreak permite saltarse algunas de las limitaciones de software impuestas por Apple en nuestros dispositivos iOS. Dependiendo de las versiones de IOS de nuestros dispositivos Apple, utilizaremos unos programas u otros. Una de las más conocidas es Pangu.
- Cliente SSH: Nos servirá para conectarnos remotamente a los dispositivos una vez el root/jailbreak esté realizado. Permite ejecutar comandos y acceder a los datos de forma remota.
- Metasploit Framework: Es una herramienta para desarrollar y ejecutar exploits contra una máquina remota, en este caso serían dispositivos móviles. Para elegir un exploit y la carga útil, se necesita un poco de información sobre el sistema objetivo, como la versión del sistema operativo y los servicios de red instalados. Esta información puede ser obtenida con el escaneo de puertos y "OS fingerprinting", a través de herramientas como Nmap.
- Smartphone Pentest Framework(SPF): Es una herramienta de seguridad de código abierto, diseñada para ayudar a evaluar la seguridad de los teléfonos inteligentes. Incluye la recopilación de información, la explotación, la ingeniería social, y la explotación posterior tanto a través de una red IP tradicional como a través de una red 2G/3G/4G, que muestra cómo pueden ser aprovechadas por los equipos de seguridad. Además permite realizar pruebas de penetración para obtener resultados de la seguridad de los teléfonos inteligentes.
- Android VTS(Vulnerability Test Suite): Esta app permite mostrar al usuario si su dispositivo móvil es susceptible de ser vulnerado, sin afectar negativamente a la estabilidad del sistema. Realiza una serie de test que comprueban una serie de errores se seguridad reconocidos en el SO Android.
- Oxygen Forensic Suite: La herramienta permite realizar un análisis forense online, es decir, con el dispositivo encendido y conectado al equipo, y extraer una amplia cantidad de información de un smartphone.

- Santoku Linux: Es una distribución de Linux basada en Lubuntu 14.04 especialmente diseñada y complementada con una serie de herramientas para llevar cabo tareas de Análisis de Malware, análisis forense de smartphones, y pruebas de seguridad de Aplicaciones. Posee una gran cantidad de herramientas ya preinstaladas y preconfiguradas para ayudarnos en futuras verificaciones y pruebas móviles. Entre ellas podemos destacar ExifTool, Iphone Backup Analyzer y SSL strip.
- Kali Linux: Es una distribución basada en Debian GNU/Linux diseñada principalmente para la auditoría y seguridad informática en general. Una de las principales virtudes de Kali Linux son las más de 300 herramientas y aplicaciones relacionadas con la seguridad informática que incluye esta distribución. Algunas aplicaciones mencionadas anteriormente las podemos encontrar en esta distro.

ANÁLISIS DE DISPOSITIVOS MÓVILES IOS

Descripción del sistema operativo iOS

iOS es el nombre del sistema operativo desarrollado por la compañía Apple Inc. exclusivo para sus dispositivos. Originalmente desarrollado para el iPhone (smartphone de Apple), aunque después se ha usado en dispositivos como el iPod touch y el iPad, que son el reproductor Mp3 y el tablet de la compañía.

Presentado en 2007 junto con el primer teléfono de la compañía, el iOS marcó una pauta sin precedentes al llegar al mercado con un sistema que no necesitaba más teclas físicas que las del volumen, encendido, bloqueo y un solitario botón llamado “Home” que permitiera al usuario a volver al inicio en su pantalla. Un sistema diseñado para ser usado con la pantalla táctil que incorporan sus dispositivos.

El sistema operativo iOS posee una interfaz fluida, sencilla y elegante, sin mucha posibilidad de personalizar pero que ofrece al usuario una de las experiencias más cómodas del mercado. Esto se debe a que iOS está diseñado para sacar el máximo provecho al hardware que de sus dispositivos, el cual siempre se ha diferenciado considerablemente de los demás fabricantes. Cuando el hardware y el software están hechos el uno para el otro las apps aprovechan al máximo elementos de hardware como el procesador, el sistema avanzado de cámaras, el sensor de huella Touch ID y la tecnología 3D Touch.

Los elementos de control consisten de deslizadores, interruptores y botones. La interacción con el sistema operativo incluye gestos como deslices, toques, pellizcos, los cuales tienen definiciones diferentes dependiendo del contexto de la interfaz. Se utilizan acelerómetros internos para hacer que algunas aplicaciones respondan a movimientos del dispositivo o rotarlo en tres dimensiones.

iOS utiliza un entorno llamado Metal para darte un rendimiento gráfico mayor. Navega por Internet, pasa de una app a otra o jugar con el juego 3D más exigente son tareas que hacen que los gráficos fluyen al máximo nivel. Con el propio sistema operativo van incluidas múltiples aplicaciones para gestionar emails, cámara, fotos, mensajes, clima, notas, contactos, calendario, reloj, etc.

Antes de la versión iOS 4 la multitarea estaba reservada para aplicaciones por defecto del sistema, y a partir del lanzamiento de dispositivos de tercera generación y de esta versión, ya se permite en otras aplicaciones.

La última versión de iOS presentada por la compañía Apple inc. es la 10.2.

El kit de desarrollo de software(SDK) fue liberado el 6 de marzo de 2008, permitiendo así a los desarrolladores hacer aplicaciones para el iPhone y iPod Touch, así como probarlas en el "iPhone simulator". De cualquier manera, y hasta la llegada de xcode 7 solo era posible utilizar el app en los dispositivos después de pagar la cuota del iPhone Developer Program. A partir de xcode 7 es posible utilizar un dispositivo iOS para probar las aplicaciones sin necesidad de tener una cuenta de desarrollador. Desde el lanzamiento de Xcode 3.1, Xcode es el programa utilizado en el iPhone SDK. Estas aplicaciones están escritas en Objective-C.

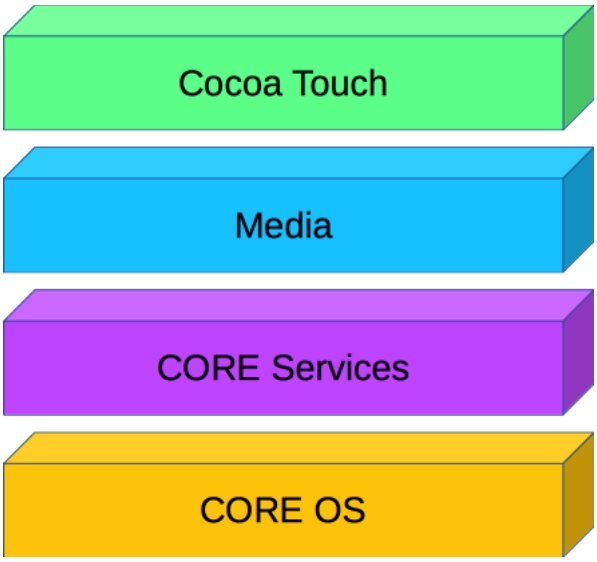
Estructura del sistema operativo iOS

El sistema operativo iOS deriva de macOS, que a su vez está basado en Darwin BSD, y por lo tanto es un sistema operativo tipo Unix.

La arquitectura de iOS se basa en capas, las capas de nivel superior interactúan como intermediarias entre el hardware y las aplicaciones que se desarrollan. Las aplicaciones no se comunican directamente con el hardware, se comunican a través de interfaces de sistemas, este mecanismo proporciona el desarrollo de aplicaciones que trabajan constantemente en dispositivos con diferentes capacidades de hardware.

Las capas de nivel superior se basan en las capas inferiores, proporcionan servicios y tecnologías más avanzadas para el desarrollo de aplicaciones, las capas inferiores poseen el control de los servicios básicos.

iOS cuenta con cuatro capas de abstracción: la capa del núcleo del sistema operativo o “Core OS”, la capa de Servicios Principales o “Core Services”, la capa de Medios de comunicaciones o “Media” y la capa de "Cocoa Touch”.



Arquitectura del sistema operativo iOS

Cada capa está compuesta por un conjunto de frameworks. A continuación describiremos cada capa en orden ascendente y sus principales características:

Core OS

Es la capa del núcleo del sistema operativo, es la capa base más baja de la pila iOS y se encuentra directamente sobre el hardware del dispositivo. Proporciona servicios de bajo nivel como:

- Gestión de memoria.

- Gestión del sistema de archivos.
- Seguridad.
- Servicios del sistema operativo.
- Gestión del acceso a red de bajo nivel.
- Gestión de procesos.
- Gestión de drivers del dispositivo.
- Control de acceso a accesorios externos.

Los drivers en esta capa proveen la interfaz entre el hardware del sistema y los frameworks del sistema. Por seguridad, el acceso al Kernel y drivers está restringido a un conjunto limitado de frameworks del sistema y aplicaciones.

iOS provee un conjunto de interfaces para el acceso a muchas características de bajo nivel del sistema operativo, a través de la biblioteca Libsystem y proveen soporte para hilos POSIX, sockets BSD API, acceso al sistema de archivos, manejo estándar de E/S, bonjour y servicios DNS, información de entorno local, asignación de memoria y cálculos matemáticos.

Core Services

Conocida también como la capa de Servicios Principales. Esta permite al usuario acceder a todos los servicios básicos y contiene los servicios fundamentales del sistema operativo que pueden ser usados por todas las aplicaciones. Muchas partes del sistema están construidas encima de esta capa.

Entre las principales tecnologías disponibles de alto nivel en esta capa se encuentran:

- Almacenamiento iCloud: permite que las aplicaciones escriban documentos y los datos a una ubicación central en Internet(espacio en la Nube) para acceder desde otros dispositivos del usuario.
- Conteo de referencias automáticas(ARC): Es una característica del compilador que simplifica la gestión de la vida útil de los objetos en Objetive C, es decir, en lugar de recordar retener o liberar un objeto el ARC evalúa las necesidades de su vida y los inserta de forma automática en las llamadas a métodos adecuados en tiempo de compilación.
- Objetos por bloque(Block Objects): Son características a nivel de lenguaje Objective-C que permiten crear segmentos de código que puede ser pasado a través de métodos o funciones como si fueran valores.
- Grand Central Dispatch(GCD): Es una tecnología de nivel BSD que se utiliza para administrar la ejecución de tareas en aplicaciones. El GCD combina un modelo de programación asincrónica con un muy optimizado núcleo para proveer la conveniencia y hacerlo más eficiente.
- Compra de APPs: Es un servicio que da la capacidad para vender contenidos y servicios desde el interior de la aplicación APP. Esta aplicación se implementa utilizando el “store kit framework” que procesa las transacciones financieras usando las cuenta de usuario iTunes.

- Biblioteca SQLite: Biblioteca que permite incrustar una base de datos ligera de SQL en aplicaciones sin ejecutar un proceso separado del servidor remoto de base de datos. Se pueden crear archivos de base de datos locales y gestionar las tablas y registros en los archivos.
- Soporte XML: La fundación framework proporciona la clase NSXMLParser para recuperación de elementos en documentos XML. Esta biblioteca de código abierto permite analizar y escribir datos XML de forma rápida y transformar el contenido XML a HTML.

Media

Esta capa contiene las tecnologías de gráficos, audio y video orientadas a crear la mejor experiencia de multimedia disponible en un dispositivo móvil. Contiene una serie de marcos que se pueden utilizar en el desarrollo de aplicaciones. Las características de las tecnologías son:

- Tecnología de gráficos: La alta calidad de los gráficos es una parte importante de las aplicaciones de iOS. Estas aplicaciones comúnmente son creadas usando el framework UIKit con vistas estándares. Sin embargo, a veces es necesario usar en aplicaciones gráficos más detalladas, para ello se puede utilizar las siguientes tecnologías para gestión del contenido gráfico como Quartz, Core animation, OpenGL y GLKit, Core Text, E/S estándar de imagen y la biblioteca Assets.
- Tecnología de audio: Las tecnologías de audio disponibles en el iOS están diseñadas para proporcionar una excelente experiencia de audio para los usuarios que incluye la capacidad de reproducir audio de alta calidad, grabar audio de alta calidad, y activar la función de vibración en determinados dispositivos. El sistema ofrece varias alternativas de reproducir y grabar contenidos de audio, existen varios tipos de tecnologías de audio de alto nivel que ofrecen flexibilidad, algunas de estas tecnologías en iOS son Media Player, Framework AV, OpenAl y el Core de audio. La tecnología de audio en iOS soporta diferentes formatos.
- Tecnología de video: iOS ofrece varias tecnologías para reproducir contenido de video en los dispositivos con hardware de video, además de tecnologías para capturar video e incorporarlo a las aplicaciones. El sistema ofrece varias formas de reproducir y grabar contenido de vídeo según necesidades. Las tecnologías de video de alto nivel simplifican el trabajo para apoyar a las características de aplicaciones por terceros. Algunas de estas tecnologías ubicadas de según nivel son la clase UIImagePickerController, el framework AV y el Core Media. Las tecnologías de vídeo en iOS son compatibles con la reproducción de archivos de vídeo con las extensiones más comunes que siguen los estándares de compresión.
- Tecnología AirPlay: AirPlay es una tecnología que permite que el flujo de audio y vídeo de aplicación a otros dispositivos compatibles como Apple TV y altavoces AirPlay de terceros a través de Wi-fi. El soporte AirPlay está integrado en el framework de AV Foundation y en la familia framework Core Audio. Cualquier contenido de audio o vídeo que reproduzca iOS se realiza automáticamente elegible para la distribución de AirPlay. Una vez usuario decide el dispositivo a sincronizar es dirigida automáticamente por el sistema.

Cocoa Touch

Es la capa principal, contiene un conjunto de Frameworks para el desarrollo de aplicaciones. Esta capa define la infraestructura de la aplicación básica y el soporte para las tecnologías punta como multitarea, entradas táctiles, notificaciones y muchos servicios de sistemas de alto nivel.

Esta interfaz provee la infraestructura básica clave para desarrollar las aplicaciones iOS, ya que contiene los recursos principales para ejecutar aplicaciones iOS, entre los recursos se distinguen las siguientes High level Features o características de alto nivel:

- Multitarea: Esta función está incorporada en la versión de iOS4 y superiores. Todas las aplicaciones que son desarrolladas y ejecutadas en el mismo sistema no terminan de ejecutarse si se pasa al escritorio o se ejecuta otra aplicación, estas aplicaciones pasan a un segundo plano o background, el cual es un contexto de ejecución de aplicaciones diferente. Esta transición desde el primer plano al segundo y viceversa es posible mediante el uso de UIKit.
- Impresión: UIKit permite enviar datos de manera inalámbrica a impresoras cercanas, esta herramienta lo hace de forma automática, por lo que el usuario solo debe darle formato a su documento.
- Protección de datos: La protección de datos permite a las aplicaciones trabajar con datos de usuario sensibles, aprovechado la encriptación implícita. Si la aplicación define un archivo como protegido, el sistema lo almacena en el disco con un formato encriptado. Cuando el dispositivo es bloqueado, el contenido de ese fichero es inaccesible, tanto para la aplicación como para cualquier potencial intruso. Cuando el dispositivo es desbloqueado, se genera una clave de desencriptación que permite a la aplicación acceder al archivo.
- Servicio de notificaciones Push: Este servicio permite notificar acerca de alguna nueva información aunque la aplicación no esté ejecutándose activamente en ese momento. Para que estas notificaciones aparezcan las aplicaciones instaladas deben solicitar la recepción de notificaciones y procesar la información una vez que ha sido recibida, además de un proceso de servidor que sea capaz de generar las notificaciones.
- Notificaciones locales: Estas notificaciones complementan el modo de notificación por push, permitiendo a la aplicación generar sus propias notificaciones sin necesidad de tener conexión con ningún servidor externo. Una vez a la notificación está programada, el sistema operativo la gestiona, por lo que la aplicación no debe ejecutarse.
- Reconocimiento de gestos: Esta característica fue introducida en la versión del sistema operativo iOS 3.2. Se trata de objetos que podemos incluir en nuestras vistas y usarlos para detectar gestos comunes, como deslizamientos o pulsaciones en la pantalla. El ulkit tiene la clase UIGestureRecognizer, que permite crear aplicaciones con este comportamiento básico para todos los gestos. Entre los gestos comunes estándares se encuentran la pulsación (Tapping), el pellizco, tanto interior como exterior, para el zoom(Pinching in and out), pulsar y arrastrar(Panning and dragging), deslizar(Swiping), rotar(Rotating) y pulsación larga(Long presses).

- **Archivos compartidos**: Se permite tener disponibles ficheros de datos del usuario en iTunes. De este modo, una aplicación hará que el contenido de su carpeta /Documents esté disponible para el usuario. Con esto, el usuario podrá añadir o quitar archivos de este directorio desde iTunes.
- **Servicios Peer-to-Peer**: El framework Game Kit permite realizar conexiones punto a punto a través de Bluetooth. Se puede usar la conectividad punto a punto para iniciar comunicaciones con dispositivos cercanos e implementar infinidad de características disponibles para juegos multijugador.
- **Controladores estándar del sistema de visitas**: Están disponibles un conjunto de view controllers para que las interfaces sean lo más estándar posibles y que el usuario se sienta más cómodo y mejore su experiencia.
- **Pantalla externa**: Está disponible la posibilidad de conectar el dispositivo a una pantalla externa a través de sus correspondientes conectores. La información acerca del dispositivo conectado está disponible mediante el framework UIKit.

SEGURIDAD EN IOS

Arquitectura de seguridad en iOS

Todos los dispositivos iOS combinan software, hardware y servicios que se han diseñado para funcionar conjuntamente con el fin de proporcionar la máxima seguridad y una experiencia de usuario transparente. iOS protege el dispositivo y los datos que contiene, así como el ecosistema en su totalidad, incluidas todas las acciones que los usuarios realizan de forma local, en redes y con servicios clave de Internet.

iOS y los dispositivos iOS proporcionan funciones de seguridad avanzadas. Muchas de estas funciones están activadas por defecto. Además las funciones de seguridad clave, como la encriptación de los dispositivos, no se pueden con configurar, y así, se evita que los usuarios las desactiven por error. Otras funciones, como Touch ID, mejoran la experiencia del usuario al facilitar la protección del dispositivo y hacerla más intuitiva.

La seguridad del sistema se ha diseñado de modo que tanto el software como el hardware estén protegidos en todos los componentes centrales de los dispositivos iOS. Esto incluye el proceso de arranque, las actualizaciones de software y el coprocesador Secure Enclave. Esta arquitectura es fundamental para la seguridad de iOS y en ningún caso interfiere en la utilización del dispositivo.

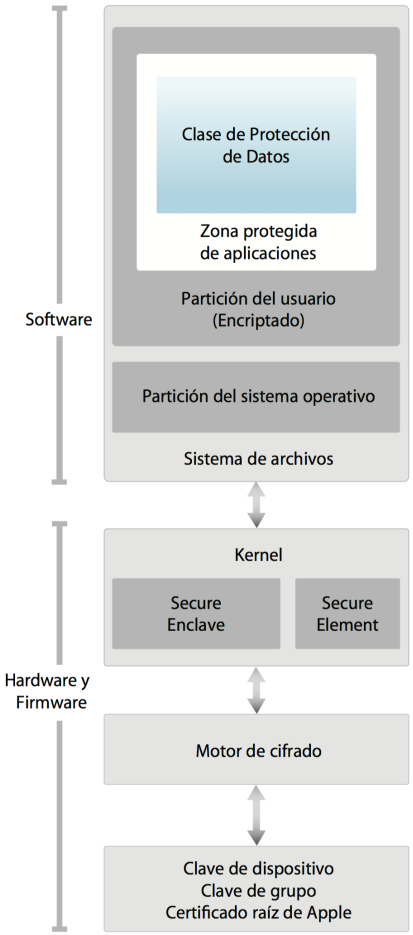
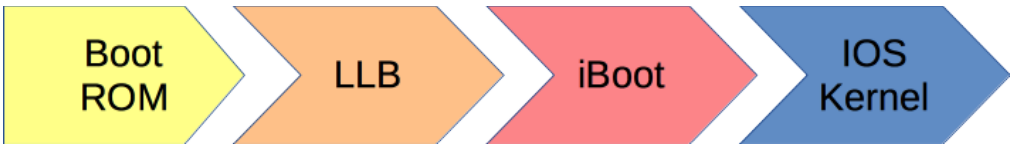


Diagrama de la arquitectura.
Fuente: www.apple.com

La estrecha integración del hardware y el software en los dispositivos iOS garantiza que todos los componentes del sistema son de confianza y valida el sistema en su conjunto. Se analizan y aprueban todos los pasos, desde el arranque inicial hasta las actualizaciones del software iOS para apps de terceros, con el fin de garantizar que el hardware y el software funcionan juntos a la perfección y utilizan los recursos correctamente.

Cadena de arranque seguro(Secure Boot Chain)

El Secure Boot Chain es el proceso de inicialización del firmware y carga de los archivos necesarios para el arranque del sistema además, se considera la capa principal del defensa para la seguridad de esta plataforma. Todos los pasos del proceso de arranque contienen componentes firmados mediante cifrado por Apple, para garantizar su integridad y el avance únicamente después de haber verificado la cadena de confianza. Esto incluye los cargadores de arranque, el kernel, las extensiones del kernel y el firmware de banda base.



Proceso de inicialización del firmware Secure Boot Chain

Cuando se enciende un dispositivo iOS, el procesador de aplicaciones ejecuta inmediatamente código de la memoria de solo lectura (o ROM de arranque). Este código inmutable, que también se conoce como raíz de confianza de hardware, se establece durante la fabricación del chip y es de confianza implícitamente. El código de la ROM de arranque contiene la clave pública de la entidad emisora de certificados (CA) raíz de Apple, que se utiliza para verificar que el cargador de arranque de bajo nivel (LLB) tiene la firma de Apple antes de permitir que se cargue. Este es el primer paso de la cadena de confianza, en la que cada paso garantiza que el siguiente está firmado por Apple. Cuando el LLB termina sus tareas, verifica y ejecuta el cargador de arranque de la siguiente fase (iBoot), que a su vez verifica y ejecuta el kernel de iOS.

La cadena de arranque seguro ayuda a garantizar que no se han manipulado los niveles de software inferiores y permite que iOS se ejecute únicamente en dispositivos Apple validados.

En el caso de los dispositivos que disponen de acceso a datos móviles, el subsistema de banda base utiliza también un proceso propio similar para el arranque seguro con software firmado y claves verificadas por el procesador de banda base. En el caso de los dispositivos que tienen un procesador A7 o uno posterior de la serie A, el coprocesador Secure Enclave utiliza también un proceso de arranque seguro que garantiza la verificación y firma de su propio software por parte de Apple.

Si un paso de este proceso de arranque no consigue cargar o verificar el siguiente proceso, se detiene el arranque y el dispositivo muestra la pantalla “Conectarse a iTunes”. Es lo que se conoce como modo de recuperación. Si la ROM de arranque no consigue cargar o verificar el LLB, entra en modo de actualización

del firmware del dispositivo (DFU). En ambos casos, el dispositivo debe conectarse a iTunes mediante USB y se deben restaurar los ajustes originales de fábrica.

Autorización del software del sistema

Apple lanza regularmente actualizaciones de software para solucionar los problemas de seguridad que van surgiendo y ofrecer nuevas características. Dichas actualizaciones se proporcionan de manera simultánea para todos los dispositivos compatibles.

Durante las actualizaciones de iOS, iTunes (o el propio dispositivo, en el caso de las actualizaciones de software OTA) se conecta al servidor de autorización de instalaciones de Apple y le envía una lista de medidas de cifrado para cada parte del paquete de instalación que se vaya a instalar (por ejemplo, el LLB, iBoot, el kernel o una imagen del sistema operativo), un valor antirreproducción aleatorio (nonce) y el identificador único del dispositivo (ECID).

El servidor de autorización coteja la lista de medidas presentada con las versiones cuya instalación se permite y, si encuentra una coincidencia, añade el ECID a la medida y firma el resultado. Como parte del proceso de actualización, el servidor envía un conjunto completo de datos firmados al dispositivo. La adición del ECID personaliza la autorización para el dispositivo que realiza la solicitud. El servidor solo autoriza y firma las medidas conocidas, de modo que se garantiza que la actualización se lleve a cabo de acuerdo con las especificaciones de Apple.

En la evaluación de la cadena de confianza durante el arranque, se verifica que la firma procede de Apple y que la medida del ítem cargado desde el disco, combinada con el ECID del dispositivo, coincide con el contenido de lo firmado.

Estos pasos garantizan que la autorización es para un dispositivo específico e impiden que una versión de iOS antigua se copie de un dispositivo a otro. El nonce impide que un atacante guarde la respuesta del servidor y la utilice para manipular un dispositivo o modificar el software del sistema de algún otro modo.

Secure Enclave

El Secure Enclave es un coprocesador incorporado en el procesador A7 o uno posterior de la serie A de Apple. Utiliza memoria encriptada e incluye un generador hardware de números aleatorios. Proporciona todas las operaciones cifradas para la gestión de claves de protección de datos y mantiene la integridad de la protección de datos, aunque el kernel haya sido comprometido. La comunicación entre el Secure Enclave y el procesador de aplicaciones está aislada en un buzón basado en interrupciones y en memorias intermedias de datos de la memoria compartida.

Durante el proceso de fabricación, se proporciona a cada Secure Enclave un identificador único (UID) propio que Apple no conoce y al que otras partes del sistema no tienen acceso. Cuando el dispositivo se enciende, se crea una clave efímera, vinculada a su UID, que se utiliza para encriptar la parte que ocupa el Secure Enclave en el espacio de la memoria del dispositivo.

Además, los datos que el Secure Enclave guarda en el sistema de archivos se encriptan con una clave vinculada al UID y un contador antirreproducciones.

Encriptación y protección de datos

iOS dispone de funciones de encriptación y de protección de datos para proteger los datos del usuario, incluso cuando otras partes de la infraestructura de seguridad están en peligro.

Todos los dispositivos iOS tienen un motor de cifrado AES de 256 bits integrado en la ruta de DMA, entre el almacenamiento flash y la memoria del sistema principal. Esto permite conseguir una encriptación de archivos muy eficiente.

Además de las funciones de encriptación de hardware integradas en los dispositivos iOS, Apple utiliza una tecnología llamada "Protección de datos" para aumentar la protección de los datos almacenados en la memoria flash del dispositivo. La protección de datos permite que el dispositivo responda ante eventos habituales, como las llamadas de teléfono entrantes, pero también permite un alto nivel de encriptación para los datos de usuario. La protección de datos se implementa mediante la creación y gestión de una jerarquía de claves, y se basa en las tecnologías de encriptación de hardware integradas en cada dispositivo iOS. La protección de datos se controla por archivo, asignando cada archivo a una clase. La accesibilidad se determina en función de si las claves de clase se han desbloqueado o no.

Seguridad de las apps

Las apps son unos de los elementos más importantes de una arquitectura moderna de seguridad de entornos móviles. Sus ventajas en cuanto a productividad son increíbles, pero si no se gestionan bien, también pueden repercutir negativamente en la seguridad y estabilidad del sistema o en los datos de usuario.

Por esta razón, iOS añade capas de protección para garantizar que las apps estén firmadas y verificadas, además de aisladas para proteger los datos de usuario.

Una vez que se ha iniciado, el kernel de iOS controla los procesos y apps del usuario que se pueden ejecutar. Para garantizar que todas las apps proceden de una fuente conocida y aprobada y no se han manipulado, iOS requiere que todo el código ejecutable se firme con un certificado emitido por Apple. Las apps proporcionadas con el dispositivo, como Mail y Safari, están firmadas por Apple. Las apps de terceros también se deben validar y firmar con un certificado emitido por Apple. La firma de código obligatoria extiende el concepto de cadena de confianza del sistema operativo a las apps e impide que aplicaciones de terceros carguen código sin firmar o utilicen código que se modifique automáticamente.

Para poder desarrollar e instalar apps en dispositivos iOS, los desarrolladores deben registrarse en Apple y unirse al programa para desarrolladores de Apple. Apple verifica la identidad real de cada desarrollador, ya sea una persona individual o una empresa, antes de emitir su certificado. Este certificado permite a los desarrolladores firmar apps y enviarlas a la tienda App Store para su distribución. Así todas las apps de App Store han sido enviadas por personas u organizaciones identificables, lo cual funciona como elemento disuasorio para la creación de apps maliciosas.

Una vez que se ha comprobado que una app procede de una fuente aprobada, iOS pone en marcha medidas de seguridad diseñadas para impedir que ponga en peligro otras apps o el resto del sistema.

Todas las apps de terceros se aíslan para impedir que accedan a los archivos almacenados por otras apps o que realicen cambios en el dispositivo. Esto evita que las apps recopilen o modifiquen la información almacenada por otras apps. Cada una tiene un directorio de inicio único para sus archivos, que se asigna de forma aleatoria al instalarla. Si una app de terceros necesita acceder a información ajena, utiliza únicamente los servicios que iOS proporciona de forma explícita.

Los archivos y recursos del sistema también están blindados contra las apps del usuario. iOS se ejecuta, mayormente, como plataforma móvil de un usuario sin privilegios, igual que todas las apps de terceros. Toda la partición del sistema operativo se instala como de solo lectura. Las herramientas innecesarias, como los servicios de inicio de sesión remoto, no se incluyen en el software del sistema y las API no permiten que las apps transfieran sus privilegios para modificar otras apps o iOS.

Seguridad de la red en iOS

Además de los métodos de protección integrados que Apple utiliza para proteger los datos almacenados en dispositivos iOS, existen muchas medidas de seguridad de la red que las organizaciones pueden poner en marcha para proteger la información durante su transferencia a un dispositivo iOS o desde él.

iOS es compatible con los protocolos de seguridad de la capa de transporte (TLS 1.0, TLS 1.1 y TLS 1.2) y DTLS. Safari, Calendario, Mail y otras apps de Internet utilizan estos mecanismos automáticamente para activar un canal de comunicación encriptado entre el dispositivo y los servicios de red. Las API de alto nivel facilitan a los desarrolladores la adopción de TLS en sus apps, mientras que las APIs de bajo nivel proporcionan un control muy preciso.

TLS

La seguridad de transporte(TLS) de las app proporciona unos requisitos de conexión por omisión, de manera que las apps cumplan las buenas prácticas para conexiones seguras. Los servidores deben ser compatibles, como mínimo, con TLS 1.2, Forward Secrecy, y los certificados deben ser válidos y estar firmados mediante SHA-256 o mejor, con un mínimo de una clave RSA de 2048 bits o una clave de curva elíptica de 256 bits.

Las conexiones de red que no cumplan estos requisitos darán error, a menos que la app omita la seguridad de transporte de las apps. Los certificados no válidos siempre dan como resultado un fallo grave e imposibilidad de conexión.

VPN

Los servicios de red segura, como las redes privadas virtuales(VPN), suelen requerir una configuración mínima para funcionar con dispositivos iOS. Estos funcionan con servidores VPN que admiten los siguientes protocolos y métodos de autenticación:

- IKEv2/IPSec con autenticación por secreto compartido, certificados RSA, certificados ECDSA, EAP-MSCHAPv2 o EAP-TLS.
- Cisco IPSec con autenticación de usuario mediante contraseña, RSA SecurID o CRYPTOCard, y autenticación de máquina mediante secreto compartido y certificados.

- L2TP/IPSec con autenticación de usuario mediante contraseña de MS-CHAPV2, RSA SecurID o CRYPTOCard, y autenticación de máquina mediante secreto compartido.
- PPTP con autenticación de usuario mediante contraseña de MS-CHAPV2 y RSA SecurID o CRYPTOCard es compatible, pero no recomendable(no disponible en iOS10).

Wi-Fi

iOS es compatible con los **protocolos Wi-Fi** estándar del sector, incluido WPA2 Enterprise, para así proporcionar acceso autenticado a redes corporativas inalámbricas. WPA2 Enterprise utiliza la encriptación AES de 128 bits para proporcionar a los usuarios la mayor garantía de que sus datos están protegidos durante las comunicaciones a través de una conexión de red Wi-Fi. Los dispositivos iOS, compatibles con 802.1X, se pueden integrar en un amplio abanico de entornos de autenticación RADIUS. El iPhone y el iPad son compatibles con los siguientes métodos de autenticación inalámbrica 802.1X: EAP-TLS, EAP-TTLS, EAP-FAST, EAP-SIM, PEAPv0, PEAPv1 y LEAP.
iOS utiliza una dirección de control de acceso al medio (MAC) aleatorizada al realizar exploraciones de preferencia de descarga de red (PNO) cuando un dispositivo no está asociado a una red Wi-Fi y su procesador está en reposo. El procesador de un dispositivo entra en reposo poco después de que se apague la pantalla.
iOS también utiliza una dirección MAC aleatorizada al realizar exploraciones de preferencia de descarga de red mejorada (ePNO) cuando un dispositivo no está asociado a una red Wi-Fi o su procesador está en reposo. Las exploraciones ePNO se ejecutan cuando un dispositivo utiliza “Localización” para apps con geocercas, como los recordatorios basados en la ubicación que determinan si el dispositivo se encuentra cerca de una ubicación específica.

Ahora la dirección MAC de un dispositivo cambia cuando no está conectado a una red Wi-Fi, por lo que no se puede utilizar para realizar un seguimiento continuo de un dispositivo con observadores pasivos del tráfico de la red Wi-Fi, incluso cuando el dispositivo está conectado a una red móvil. Las exploraciones en segundo plano utilizan una dirección MAC aleatorizada, y que ni Apple ni los fabricantes pueden predecir estas direcciones MAC aleatorias. La aleatorización de las direcciones MAC Wi-Fi no es compatible con iPhone 4s.

BlueTooth

La conectividad Bluetooth en iOS se ha diseñado de modo que su funcionalidad resulte útil y que el acceso a datos privados no aumente innecesariamente. Los dispositivos iOS admiten conexiones Encryption Mode 3, Security Mode 4 y Service Level 1. La compatibilidad varía en función del dispositivo.

Seguridad de AirDrop

Los dispositivos iOS compatibles con AirDrop utilizan Bluetooth de baja energía (BLE o Bluetooth LE) y la tecnología Wi-Fi P2P creada por Apple para enviar archivos e información a dispositivos cercanos.

Cuando un usuario activa AirDrop, se almacena una identidad RSA de 2048 bits en el dispositivo. Además, se crea un hash de identidad de AirDrop basado en las direcciones de correo electrónico y los números de teléfono asociados al ID de Apple del usuario.

Cuando un usuario elige AirDrop como método para compartir un ítem, el dispositivo emite una señal de AirDrop a través de Bluetooth LE. Los dispositivos que estén activos, se encuentren cerca y tengan AirDrop activado detectarán la señal y responderán con una versión abreviada del hash de identidad de su propietario.

Seguridad de Apple Pay con NFC

Con Apple Pay, los usuarios pueden utilizar el Apple Watch y los dispositivos iOS compatibles para pagar de forma sencilla, segura y privada. Es un sistema fácil para los usuarios que incluye seguridad integrada tanto en el hardware como en el software.

Además Apple Pay se ha diseñado para proteger la información personal del usuario. Apple Pay no recopila información de las transacciones que se pueda vincular al usuario. Las transacciones de pago quedan entre el usuario, el beneficiario y la entidad emisora de la tarjeta. Los componentes más importantes son:

Secure Element: El Secure Element es un chip certificado estándar del sector que ejecuta la plataforma Java Card. Esta plataforma cumple los requisitos del sector financiero en cuanto a pagos electrónicos.

Controlador NFC: El controlador de comunicación de corto alcance (NFC) gestiona los protocolos NFC y dirige la comunicación entre el procesador de aplicaciones y el Secure Element, y entre el Secure Element y el terminal del punto de venta.

Secure Element aloja un applet diseñado específicamente para gestionar Apple Pay. También incluye applets de pago certificados por las redes de pago. Los datos de las tarjetas de crédito o débito se envían a estos applets de pago desde la red de pago o la entidad emisora de la tarjeta, encriptados con claves que solo conocen la red de pago y el dominio de seguridad de los applets de pago. Estos datos se almacenan en los applets de pago y se protegen con las funciones de seguridad del Secure Element. Durante una transacción, el terminal se comunica directamente con el Secure Element a través del controlador NFC mediante un bus de hardware dedicado.

Como puerta de enlace al Secure Element, el controlador NFC garantiza que todas las transacciones de pago sin contacto se realizan a través de un terminal del punto de venta que esté cerca del dispositivo. El controlador NFC solo marca como transacciones sin contacto aquellas solicitudes de pago procedentes de un terminal del área. Una vez que el titular de la tarjeta autoriza el pago mediante Touch ID o su código, o bien al hacer doble clic en el botón lateral de un Apple Watch desbloqueado, el controlador dirige las respuestas sin contacto preparadas por los applets de pago del Secure Element al campo de NFC de forma exclusiva. En consecuencia, los datos de autorización de pagos para las transacciones sin contacto se incluyen en el campo local de NFC y nunca se exponen al procesador de aplicaciones. En comparación, los datos de autorización de pagos realizados en las apps se dirigen al procesador de aplicaciones, pero siempre después de que el Secure Element los encripte en el servidor de Apple Pay.

Servicios de Internet

Apple ha creado un conjunto de servicios para ayudar a los usuarios a aprovechar la utilidad y productividad de sus dispositivos. Estos servicios incluyen iMessage, FaceTime, Siri, sugerencias de Spotlight, iCloud, copia de seguridad de iCloud y llavero de iCloud.

Estos servicios de Internet se han diseñado con los mismos objetivos de seguridad que iOS promueve en toda su plataforma. Dichos objetivos incluyen la gestión segura de datos, tanto si no se están utilizando en el dispositivo como si se están transfiriendo por redes inalámbricas; la protección de la información personal de los usuarios; y la protección frente al acceso malintencionado o no autorizado a la información y los servicios. Cada servicio utiliza su propia arquitectura de seguridad sin comprometer la facilidad de uso global de iOS.

ID de Apple

El ID de Apple está constituido por el nombre y la contraseña del usuario necesarios para iniciar sesión en servicios de Apple. Es importante que el usuario proteja su ID de Apple para evitar que se produzca un acceso no autorizado a sus cuentas. Con el fin de ayudarle a conseguirlo, Apple exige el uso de contraseñas seguras compuestas, al menos, de ocho caracteres que combinen números y letras, que no contengan el mismo carácter repetido más de tres veces de forma consecutiva y que no sean de uso común. Se recomienda a los usuarios que aumenten el grado de protección indicado añadiendo más caracteres o signos de puntuación para que sus contraseñas resulten aún más seguras. Apple también requiere a los usuarios que configuren tres preguntas de seguridad que se puedan utilizar para ayudar a comprobar la identidad del propietario a la hora de realizar cambios en la información de su cuenta o de restablecer una contraseña olvidada.

Apple también envía mensajes de correo electrónico y notificaciones push a los usuarios cuando se producen cambios importantes en sus cuentas. Además, Apple utiliza diversas políticas y procedimientos diseñados para proteger las cuentas de los usuarios. Entre ellos, se incluyen la limitación del número de veces que se puede intentar iniciar sesión y restablecer la contraseña, la supervisión activa de fraudes para ayudar a identificar los ataques mientras se están produciendo, y las revisiones periódicas de las políticas que permiten adaptarnos a cualquier información nueva que pueda afectar a la seguridad del cliente.

Para ayudar a los usuarios a proteger más sus cuentas, Apple ofrece la autenticación de doble factor. Este sistema de autenticación es una capa adicional de seguridad para los ID de Apple. Está diseñado para garantizar que solo el propietario de la cuenta pueda acceder a ella, aunque otra persona conozca la contraseña. Para iniciar sesión por primera vez en un dispositivo nuevo, se necesita la contraseña del ID de Apple y un código de verificación de seis dígitos que se muestra en los dispositivos de confianza del usuario o que se envía a un número de teléfono de confianza automáticamente. Al introducir el código, el usuario verifica que el dispositivo nuevo es de confianza y que es seguro iniciar sesión en él. Puesto que para acceder a la cuenta de un usuario, ya no basta con una contraseña, la autenticación de doble factor mejora la seguridad del ID de Apple del usuario y de toda la información personal que este guarda en Apple.

Desde 2013, Apple también ofrece un método de seguridad parecido denominado verificación en dos pasos. Con este método activado, la identidad del usuario se debe comprobar mediante un código temporal que se envía a uno de los dispositivos de confianza del usuario antes de permitir que se modifique la información de

la cuenta de su ID de Apple; antes de iniciar sesión en iCloud, iMessage, FaceTime y Game Center; o antes de realizar compras en iTunes Store, iBooks Store o App Store desde un dispositivo nuevo. Los usuarios también reciben una clave de recuperación de 14 caracteres que deben guardar en un lugar seguro para usarla en caso de olvidar su contraseña o perder el acceso a los dispositivos de confianza.

iMessage

iMessage de Apple es un servicio de mensajería para dispositivos iOS y ordenadores Mac. iMessage admite texto y archivos adjuntos tales como fotos, contactos y ubicaciones. Puesto que los mensajes se muestran en todos los dispositivos registrados de un usuario, una conversación se puede continuar desde cualquiera de sus dispositivos. iMessage utiliza el servicio de notificaciones push de Apple (APNs) en gran medida. Apple no registra mensajes ni archivos adjuntos y su contenido está protegido mediante una encriptación de punto a punto, de modo que únicamente el emisor y el receptor pueden acceder a ellos, ya que Apple no puede desencriptar los datos.

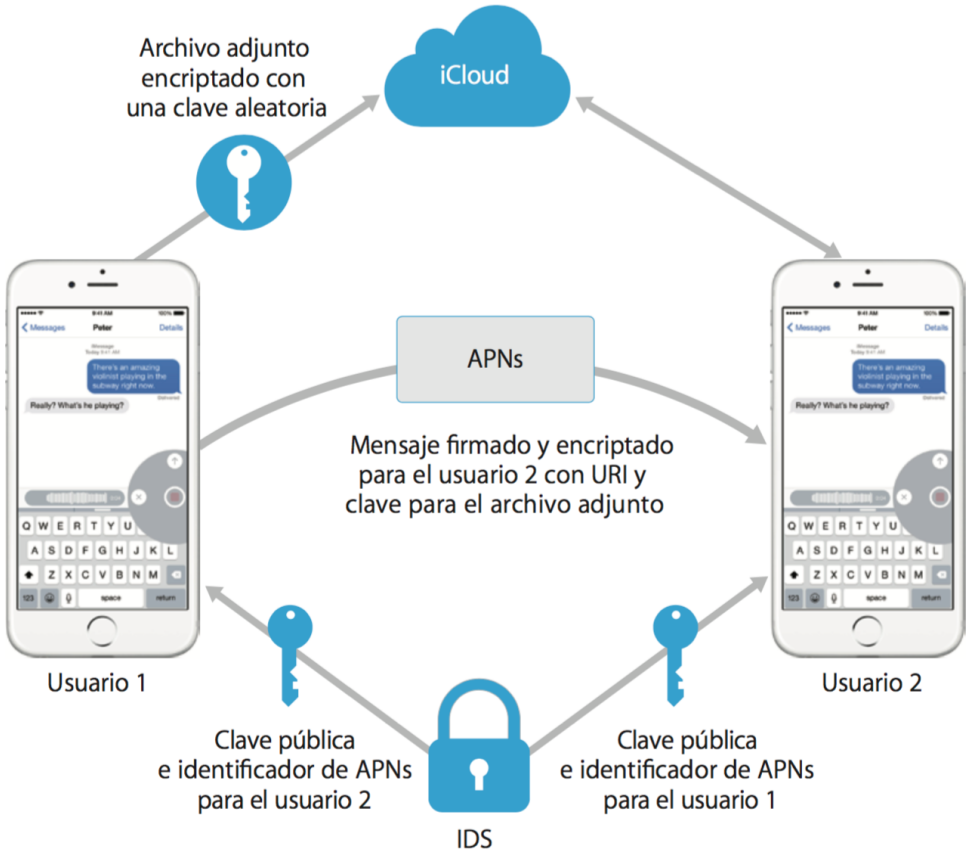


Diagrama del envío de mensajes con iMessage. Fuente: www.apple.com

Cuando un usuario activa iMessage en un dispositivo, el dispositivo genera dos pares de claves para usarlas con el servicio: una clave RSA de 1280 bits para la encriptación y una clave ECDSA de 256 bits en la curva

P-256 del NIST para el inicio de sesión. Las claves privadas de ambos pares de claves se guardan en el llavero del dispositivo y las claves públicas se envían al servicio de directorio (IDS) de Apple, donde se asocian al número de teléfono o la dirección de correo electrónico del usuario, junto con la dirección del APNs del dispositivo. Los usuarios inician una nueva conversación de iMessage al introducir una dirección o un nombre, el dispositivo utiliza primero la app Contactos del usuario para recopilar los números de teléfono y las direcciones de correo electrónico asociadas a ese nombre y, a continuación, obtiene las claves públicas y las direcciones del APNs del IDS de Apple. El mensaje que envía el usuario está encriptado de forma individual para cada uno de los dispositivos del destinatario. Las claves de encriptación RSA públicas de los dispositivos receptores se obtienen del IDS. Para cada dispositivo receptor, el dispositivo emisor genera una clave aleatoria de 128 bits que utiliza para encriptar el mensaje con AES en modo CTR. Esta clave AES por mensaje se encripta con RSA-OAEP para la clave pública del dispositivo receptor. Con el texto del mensaje encriptado y la clave del mensaje encriptada se genera un hash SHA-1, que se firma con ECDSA utilizando la clave de firma privada del dispositivo emisor. Los mensajes que se obtienen, uno para cada dispositivo receptor, están constituidos por el texto del mensaje encriptado, la clave del mensaje encriptada y la firma digital del emisor. A continuación, se mandan al APNs para que los envíe. Los metadatos, como la fecha y la información sobre el enrutamiento del APNs no se encriptan. La comunicación con el APNs se encripta utilizando un canal TLS de secreto-hacia-delante.

El APNs solo puede transmitir mensajes de 4 o 16 KB como máximo en función de la versión de iOS. Si el texto del mensaje es demasiado largo o si se incluye un archivo adjunto (por ejemplo, una foto), el archivo adjunto se encripta con AES en modo CTR utilizando una clave de 256 bits generada aleatoriamente y se carga a iCloud. A continuación, la clave AES para el archivo adjunto, su identificador de recursos uniforme (URI) y un hash SHA-1 de su forma encriptada se envían al destinatario como el contenido de un mensaje de iMessage, cuya confidencialidad e integridad están protegidas mediante la encriptación normal de iMessage. En cuanto a la recepción, cada dispositivo recibe una copia del mensaje desde el APNs y, en caso necesario, recupera el archivo adjunto de iCloud. Como sucede con todas las notificaciones push, el mensaje se elimina del APNs una vez enviado. Sin embargo, a diferencia de lo que sucede con otras notificaciones del APNs, los mensajes de iMessage se ponen en la cola para enviarlos a los dispositivos sin conexión. Actualmente, los mensajes se almacenan durante un plazo máximo de 30 días.

FaceTime

FaceTime es el servicio de llamadas de audio y vídeo de Apple. De forma parecida a iMessage, las llamadas de FaceTime también utilizan el servicio de notificaciones push de Apple para establecer una conexión inicial con los dispositivos registrados del usuario. El contenido de audio/vídeo de las llamadas FaceTime se protege mediante la encriptación de punto a punto, con lo cual, únicamente el emisor y el receptor pueden acceder a él ya que Apple no puede desencriptar los datos.

FaceTime utiliza el establecimiento de conectividad de Internet (ICE) para establecer una conexión P2P entre los dispositivos. Al usar mensajes conforme al Protocolo de inicio de sesión (SIP), los dispositivos verifican sus certificados de identidad y establecen un secreto compartido para cada sesión. Los *nonces* cifrados que suministra cada dispositivo se combinan con claves de sal para cada uno de los canales de contenido

multimedia, que se transmiten mediante el Protocolo en tiempo real seguro (SRTP) con la encriptación AES-256.

iCloud

iCloud almacena los contactos, calendarios, fotos, documentos y otra información del usuario, y la mantiene al día automáticamente en todos sus dispositivos. Además, las apps de terceros también pueden usar iCloud para almacenar y sincronizar documentos, así como datos clave-valor para datos de apps según las indicaciones del desarrollador. Los usuarios configuran iCloud al iniciar sesión con un ID de Apple y seleccionar qué servicios desean usar.

iCloud encripta cada archivo, que se desglosa en fragmentos, con AES-128 y una clave derivada del contenido de cada fragmento que utiliza SHA-256. Apple almacena las claves y los metadatos de los archivos en la cuenta de iCloud del usuario. Los fragmentos encriptados del archivo se almacenan (sin incluir información identificativa del usuario) mediante servicios de almacenamiento de terceros como Amazon S3 y Microsoft Azure.

Copias de seguridad de iCloud

iCloud también realiza copias de seguridad de información (como ajustes de los dispositivos, datos de las apps, fotos y vídeos en Carrete, así como conversaciones en la app Mensajes) a diario y a través de la red Wi-Fi. Para proteger el contenido, iCloud lo encripta cuando se envía por Internet, lo almacena en un formato encriptado y utiliza identificadores seguros para su autenticación. Las copias de seguridad de iCloud se llevan a cabo únicamente cuando el dispositivo está bloqueado, conectado a una fuente de alimentación y con acceso a Internet mediante Wi-Fi. Debido a la encriptación que se usa en iOS, el sistema se ha diseñado para mantener protegidos los datos y, al mismo tiempo, permitir que se realicen copias de seguridad y restauraciones progresivas y sin supervisión.

El conjunto de copias de seguridad se almacena en la cuenta de iCloud del usuario y consiste en una copia de los archivos del usuario y el repositorio de claves de copia de seguridad de iCloud. Este repositorio está protegido mediante una clave aleatoria, que también está almacenada en el conjunto de copias de seguridad. (La contraseña de iCloud del usuario no se usa para la encriptación. De este modo, el cambio de contraseña de iCloud no invalida las copias de seguridad existentes.

Mientras se mantenga una copia de la base de datos del llavero del usuario en iCloud, permanecerá protegida mediante una clave vinculada al UID. Esto permite que el llave- ro solo se pueda restaurar en el mismo dispositivo en el que se originó, es decir, nadie (incluido Apple) podrá leer los ítems del llavero del usuario.

Al restaurarlo, los archivos de los que se ha realizado la copia de seguridad, el repositorio de claves de copia de seguridad de iCloud y la clave para el repositorio de claves se recuperan de la cuenta de iCloud del usuario. El repositorio de claves de copia de seguridad de iCloud se desencripta usando su clave, a continuación, las claves por archivo del repositorio de claves se usan para desencriptar los archivos del conjunto de copias de seguridad, que se escriben como archivos nuevos en el sistema de archivos y, de este modo, se vuelven a encriptar según la clase de protección de datos correspondiente.

Llavero de iCloud

El llavero de iCloud permite a los usuarios sincronizar de forma segura sus contraseñas entre dispositivos iOS y ordenadores Mac sin exponer esa información a Apple. Cuando un usuario activa el llavero de iCloud por primera vez, el dispositivo establece un círculo de confianza y crea una identidad de sincronización para sí mismo. La identidad de sincronización consta de una clave privada y una clave pública. La clave pública de la identidad de sincronización se coloca en el círculo y el círculo se firma dos veces: primero, lo firma la clave privada de la identidad de sincronización y, después, una clave de curva elíptica asimétrica (con P256) derivada de la contraseña de la cuenta de iCloud del usuario. Junto con el círculo, se almacenan los parámetros (sal aleatoria e iteraciones), usados para crear la clave que se basa en la contraseña de iCloud del usuario.

El círculo de sincronización firmado se ubica en el área de almacenamiento de datos clave-valor de iCloud del usuario. No se puede leer sin conocer la contraseña de iCloud del usuario y no se puede modificar de forma válida sin disponer de la clave privada de la identidad de sincronización de su miembro. Cuando el usuario activa el llavero de iCloud en otro dispositivo, el nuevo dispositivo detecta en iCloud que el usuario dispone de un círculo de sincronización previo establecido al que no pertenece. El dispositivo crea el par de claves de identidad de sincronización correspondiente y, a continuación, crea un vale de aplicación para solicitar formar parte de ese círculo como miembro. El vale consta de la clave pública del dispositivo de su identidad de sincronización y se solicita al usuario que se autentique con su contraseña de iCloud. Los parámetros de generación de la clave de curva elíptica se recuperan de iCloud y se genera una clave que se usa para firmar el vale de aplicación. Por último, este vale se coloca en iCloud.

Cuando el primer dispositivo detecta la recepción de un vale de aplicación, muestra un aviso para que el usuario sepa que hay un dispositivo nuevo que solicita entrar en el círculo de sincronización. El usuario introduce su contraseña de iCloud, y se comprueba que el vale de aplicación está firmado por una clave privada coincidente. Esto determina que la persona que ha generado la solicitud para entrar en el círculo ha introducido la contraseña de iCloud del usuario cuando se le ha solicitado.

Tras la aprobación del usuario para añadir un dispositivo nuevo al círculo, el primer dispositivo añade la clave pública del nuevo miembro al círculo de sincronización, vuelve a firmarlo con la identidad de sincronización correspondiente y la clave derivada de la contraseña de iCloud del usuario. El nuevo círculo de sincronización se ubica en iCloud, donde el nuevo miembro lo firma de manera similar.

Así, el círculo de sincronización tiene dos miembros y cada uno de ellos dispone de la clave pública del otro. Estos empiezan a intercambiar ítems individuales del llavero mediante el almacenamiento de datos clave-valor de iCloud.

Siri

Siri es una aplicación con funciones de asistente personal para iOS. Esta aplicación utiliza procesamiento del lenguaje natural para responder preguntas, hacer recomendaciones y realizar acciones mediante la delegación de solicitudes hacia un conjunto de servicios web que ha ido aumentando con el tiempo.

Los usuarios pueden utilizar Siri para enviar mensajes, organizar reuniones y hacer llamadas telefónicas, entre otras cosas, hablándole de forma natural. Siri utiliza el reconocimiento de voz, la conversión de texto a voz y un modelo cliente-servidor para responder a una amplia variedad de solicitudes. Las tareas que Siri admite se han diseñado para garantizar que solamente se utiliza la cantidad mínima de información personal y que está completamente protegida.

Cuando Siri se activa, el dispositivo crea identificadores aleatorios para usar con el reconocimiento de voz y los servidores de Siri. Estos identificadores se usan únicamente dentro de Siri y sirven para mejorar el servicio. Si después Siri se desactiva, el dispositivo genera un identificador aleatorio nuevo para usarlo cuando se vuelva a activar.

Handoff

Continuidad saca provecho de tecnologías como iCloud, Bluetooth y Wi-Fi para permitir a los usuarios continuar con una actividad en otro dispositivo, hacer y recibir llama- das telefónicas, enviar y recibir mensajes de texto, así como compartir la conexión a Internet de un dispositivo móvil.

Con Handoff, cuando el Mac y el dispositivo iOS de un usuario están cerca, el usuario puede transferir automáticamente aquello en lo que esté trabajando de un dispositivo al otro. Handoff permite al usuario cambiar de dispositivo y continuar trabajando de forma instantánea.

Cuando un usuario inicia sesión en iCloud en un segundo dispositivo compatible con Handoff , los dos dispositivos establecen un enlace mediante una conexión Bluetooth LE 4.0 fuera de banda a través del APNs. Los mensajes individuales están encriptados de forma similar a como sucede en iMessage. Una vez que los dispositivos están enlazados, cada uno genera una clave simétrica AES de 256 bits que se almacena en el llavero del dispositivo. Esta clave se usa para encriptar y autenticar los avisos de la conexión Bluetooth LE que comunican la actividad actual del dispositivo con otros dispositivos enlazados de iCloud utilizando AES-256 en el modo GCM con medidas de protección de reproducción. La primera vez que un dispositivo recibe un aviso de una clave nueva, establece una conexión Bluetooth LE con el dispositivo que origina la clave y genera un intercambio de claves de encriptación del aviso. Esta conexión se protege mediante la encriptación estándar Bluetooth LE 4.0, así como mediante la encriptación de los mensajes individuales, que es parecida a la encriptación de iMessage. En algunas situaciones, estos mensajes se envían mediante el servicio de notificaciones push de Apple en lugar de mediante la conexión Bluetooth LE. La carga útil de la actividad se protege y se transfiere del mismo modo que con un iMessage.

ANÁLISIS DE DISPOSITIVOS MÓVILES ANDROID

Descripción del sistema operativo Android

Android es un sistema operativo orientado a dispositivos móviles, basado en una versión modificada del núcleo Linux. Inicialmente fue desarrollado por Android Inc., una pequeña empresa, que posteriormente fue comprada por Google; en la actualidad lo desarrollan los miembros de la Open Handset Alliance (liderada por Google).

Su presentación se realizó el 5 de noviembre de 2007 junto con la fundación Open Handset Alliance, en un consorcio de numerosas compañías de hardware, software y telecomunicaciones comprometidas con la promoción de estándares abiertos para dispositivos móviles. El primer teléfono disponible en el mercado para ejecutar Android fue el HTC Dream, dado a conocer al público el 22 de octubre de 2008.

La plataforma de hardware principal de Android es la arquitectura ARM, aunque también hay soporte para x86 en el proyecto Android-x86.

El sistema operativo Android es un sistema abierto, multitarea, que permite a los desarrolladores acceder a las funcionalidades principales del dispositivo mediante aplicaciones, cualquier aplicación puede ser reemplazada libremente, además desarrollarlas por terceros, a través de herramientas proporcionadas por Google, y mediante los lenguajes de programación Java y C. El sistema operativo proporciona todas las interfaces necesarias para desarrollar aplicaciones que accedan a las funciones del teléfono (como el GPS, las llamadas, la agenda, etc.) de una forma muy sencilla.

El código fuente de Android está disponible bajo diversas licencias de software libre y código abierto, Google liberó la mayoría del código de Android bajo la licencia Apache. Todo esto permite que un desarrollador no solo pueda modificar su código sino también mejorarlo. A través de esas mejoras puede publicar el nuevo código y con el ayudar a mejorar el sistema operativo para futuras versiones.

Android depende de un Linux versión 2.6 para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, stack de red, y modelo de drivers. El núcleo también actúa como una capa de abstracción entre el hardware y el resto del stack de software.

Otras características que presenta Android es un navegador web integrado basado en el motor WebKit, soporte para gráfico 2D y 3D basado en la especificación OpenGL, soporte multimedia para audio, video e imágenes en varios formatos, conectividad Bluetooth, EDGE, 3G, Wifi, entre otros.

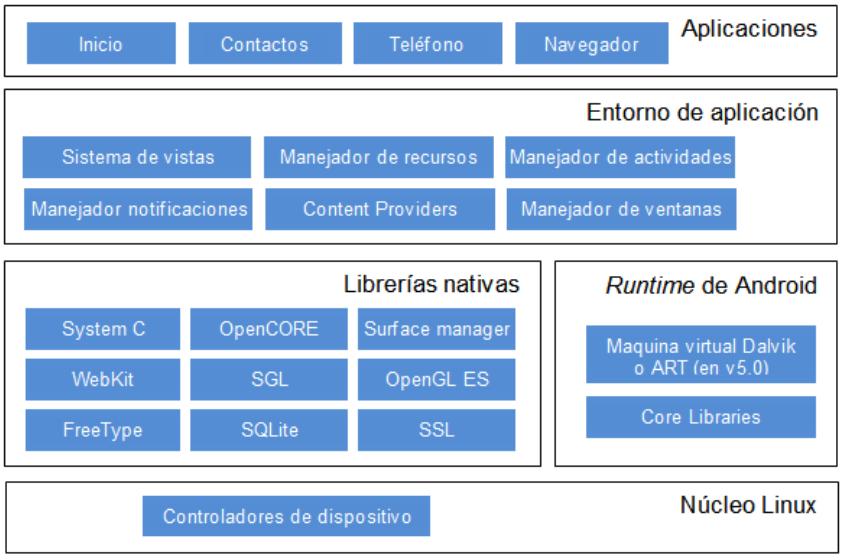
Cabe resaltar, que la libertad del código de Android ha logrado que en poco tiempo se implante en multitud de dispositivos electrónicos, desde móviles hasta computadoras portátiles, netbooks, microondas, lavadoras, marcos digitales, navegadores GPS, relojes e incluso en navegadores de abordo de coches. Esto convierte a Android en un sistema operativo multifuncional, que garantiza cada vez más su crecimiento y expansión en el mercado y fabricación tecnológica.

Las versiones de Android reciben, en inglés, el nombre de diferentes postres o dulces. En cada versión el postre o dulce elegido empieza por una letra distinta, conforme a un orden alfabético. La última versión disponible de Android es la 7.0, que corresponde a la letra N proveniente de Nougat.

Estructura del sistema operativo Android

Android está formada por varias capas que facilitan al desarrollador la creación de aplicaciones. Además, esta distribución permite acceder a las capas más bajas mediante el uso de librerías, facilitando las tareas del desarrollador y así no precise programar a bajo nivel las funcionalidades necesarias para que una aplicación haga uso de los componentes de hardware del smartphone. Cada una de las capas utiliza elementos de la capa inferior para realizar sus funciones, es por ello que a este tipo de arquitectura se le conoce también como pila.

La arquitectura de Android está formada por cinco capas. Una de las características más importantes es que todas las capas están basadas en software libre.



Arquitectura del sistema operativo Android. Fuente: www.andoridcurso.com

Núcleo Linux

El núcleo del sistema operativo Android está basado en el kernel de Linux versión 2.6, solo que adaptado a las características del hardware en el que se ejecutará Android, es decir, adaptado al hardware de dispositivos móviles.

El núcleo actúa como una capa de abstracción entre el hardware y el resto de las capas de la arquitectura. Esto permite que se pueda acceder a esos componentes sin necesidad de conocer el modelo o características precisas de los que están instalados en cada teléfono. Para cada elemento hardware del teléfono existe un controlador dentro del kernel que permite utilizarlo desde el software.

El kernel también se encarga de gestionar los diferentes recursos del teléfono (energía, memoria, etc.) y del sistema operativo en sí: procesos, elementos de comunicación (networking), etc.

Runtime de Android

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

Cabe aclarar que Dalvik es una variación de la máquina virtual de Java, por lo que no es compatible con el bytecode Java. Java se usa únicamente como lenguaje de programación, y los ejecutables que se generan con el SDK de Android tienen la extensión .dex que es específico para Dalvik, y por ello no podemos correr aplicaciones Java en Android ni viceversa. A partir de Android 5.0 se reemplaza Dalvik por ART. Esta nueva máquina virtual consigue reducir el tiempo de ejecución del código Java hasta en un 33%.

También se incluye en el runtime de Android el módulo Core Libraries, con la mayoría de las librerías disponibles en el lenguaje Java.

Bibliotecas nativas

La capa que se sitúa justo sobre el kernel la componen las bibliotecas nativas de Android. Estas bibliotecas están escritas en C o C++ y compiladas para la arquitectura hardware específica del teléfono. Su cometido es proporcionar funcionalidad a las aplicaciones, para tareas que se repiten con frecuencia, evitando tener que codificarlas cada vez y garantizando que se llevan a cabo de la forma más eficiente. Están compiladas en código nativo del procesador. Muchas de las librerías utilizan proyectos de código abierto. Algunas de estas librerías son:

- System C library: una derivación de la librería BSD de C estándar (libc), adaptada para dispositivos embebidos basados en Linux.
- Media Framework: librería basada en OpenCORE de PacketVideo. Soporta codecs de reproducción y grabación de multitud de formatos de audio y vídeo e imágenes MPEG4, H.264, MP3, AAC, AMR, JPG y PNG.
- Surface Manager: maneja el acceso al subsistema de representación gráfica en 2D y 3D.
- WebKit/Chromium: soporta un moderno navegador Web utilizado en el navegador Android y en la vista Webview. En la versión 4.4, WebKit ha sido reemplazada por Chromium/Blink, que es la base del navegador Chrome de Google.
- SGL: motor de gráficos 2D.
- Librerías 3D: implementación basada en OpenGL ES 1.0 API. Las librerías utilizan el acelerador harware 3D si está disponible, o el software altamente optimizado de proyección 3D.
- FreeType: fuentes en bitmap y renderizado vectorial.

- SQLite: potente y ligero motor de bases de datos relacionales disponible para todas las aplicaciones.
- SSL: proporciona servicios de encriptación Secure Socket Layer (capa de conexión segura).

Entorno de aplicación

La siguiente capa la forman todas las clases y servicios que utilizan directamente las aplicaciones para realizar sus funciones y que, obviamente, se apoyan en las bibliotecas y en el entorno de ejecución que ya hemos detallado. La mayoría de los componentes de esta capa son bibliotecas Java que acceden a los recursos a través de la máquina virtual Dalvik. Entre las más importantes se encuentran las siguientes:

- Administrador de actividades (Activity Manager): se encarga de controlar el ciclo de vida de las actividades (del que hablo en otro post) y la propia pila de actividades. Sin entrar en muchos detalles ahora, las actividades se pueden definir como las ventanas que se muestran, una sobre otra, en la pantalla del dispositivo Android (usando un concepto de ventana similar al de los sistemas operativos gráficos de PC, aunque el funcionamiento en Android sea muy diferente).
- Administrador de ventanas (Windows Manager): se encarga de organizar lo que se muestra en pantalla, creando superficies que pueden ser rellenadas por las actividades.
- Proveedor de contenidos (Content Provider): permite encapsular un conjunto de datos que va a ser compartido entre aplicaciones creando una capa de abstracción que hace accesible dichos datos sin perder el control sobre cómo se accede a la información. Por ejemplo, uno de los proveedores de contenido existentes permite a las aplicaciones acceder a los contactos almacenados en el teléfono. Esta biblioteca nos permite crear también nuestros propios proveedores para permitir que otras aplicaciones accedan a información que gestiona la nuestra.
- Vistas (Views): si antes equiparábamos las actividades con las ventanas de un sistema operativo de PC, las vistas las podríamos equiparar con los controles que se suelen incluir dentro de esas ventanas. Android proporciona numerosas vistas con las que construir las interfaces de usuario: botones, cuadros de texto, listas, etc. También proporciona otras más sofisticadas, como un navegador web o un visor de Google Maps.
- Administrador de notificaciones (Notification Manager): proporciona servicios para notificar al usuario cuando algo requiera su atención. Normalmente las notificaciones se realizan mostrando alerta en la barra de estado, pero esta biblioteca también permite emitir sonidos, activar el vibrador o hacer pardear los LEDs del teléfono (si los tiene).
- Administrador de paquetes (Package Manager): las aplicaciones Android se distribuyen en paquetes (archivos .apk) que contienen tanto los archivos .dex como todos los recursos y archivos adicionales que necesite la aplicación, para facilitar su descarga e instalación. Esta biblioteca permite obtener información sobre los paquetes actualmente instalados en el dispositivo Android, además de gestionar la instalación de nuevos paquetes.
- Administrador de telefonía (Telephony Manager): proporciona acceso a la pila hardware de telefonía del dispositivo Android, si la tiene. Permite realizar llamadas o enviar y recibir SMS/MMS, aunque no permite

reemplazar o eliminar la actividad que se muestra cuando una llamada está en curso (por motivos de seguridad).

- Administrador de recursos (Resource Manager): proporciona acceso a todos los elementos propios de una aplicación que se incluyen directamente en el código: cadenas de texto traducidas a diferentes idiomas, imágenes, sonidos e incluso disposiciones de las vistas dentro de una actividad (layouts). Permite gestionar esos elementos fuera del código de la aplicación y proporcionar diferentes versiones en función del idioma del dispositivo o la resolución de pantalla que tenga, por ejemplo.
- Administrador de ubicaciones (Location Manager): permite determinar la posición geográfica del dispositivo Android (usando el GPS o las redes disponibles) y trabajar con mapas.
- Administrador de sensores (Sensor Manager): permite gestionar todos los sensores hardware disponibles en el dispositivo Android: acelerómetro, giroscopio, sensor de luminosidad, sensor de campo magnético, brújula, sensor de presión, sensor de proximidad, sensor de temperatura, etc.
- Cámara: proporciona acceso a las cámaras del dispositivo Android, tanto para tomar fotografías como para grabar vídeo.
- Multimedia: conjunto de bibliotecas que permiten reproducir y visualizar audio, vídeo e imágenes en el dispositivo.

Aplicaciones

La capa superior de la arquitectura Android la forman, como no podría ser de otra forma, las aplicaciones. En esta capa se incluyen todas las aplicaciones del dispositivo, tanto las que tienen interfaz de usuario como las que no, tanto las nativas (programadas en C o C++) como las administradas (programadas en Java), tanto las que vienen de serie con el dispositivo como las instaladas por el usuario.

Aquí está también la aplicación principal del sistema: Inicio (Home), también llamada a veces lanzador (launcher), porque es la que permite ejecutar otras aplicaciones proporcionando la lista de aplicaciones instaladas y mostrando diferentes escritorios donde se pueden colocar accesos directos a aplicaciones o incluso pequeñas aplicaciones incrustadas o widgets, que son también aplicaciones de esta capa.

Lo principal a tener en cuenta de esta arquitectura es que todas las aplicaciones, ya sean las nativas de Android, las que proporciona Google, las que incluye de serie el fabricante del teléfono o las que instala después el usuario utilizan el mismo marco de aplicación para acceder a los servicios que proporciona el sistema operativo. Esto implica dos cosas: que podemos crear aplicaciones que usen los mismos recursos que usan las aplicaciones nativas (nada está reservado o inaccesible) y que podemos reemplazar cualquiera de las aplicaciones del teléfono por otra de nuestra elección. En Android se tiene el control total por parte del usuario del software que se ejecuta en su teléfono.

Sistema de archivos Android

Después de dar un enfoque general a la arquitectura de Android, se entiende que es un Sistema Operativo basado Linux, su estructura de sistema de archivos no es una excepción. En Android hay disponibles 2 sistemas EXT4 y F2FS.

En el caso de EXT4, hablamos del sistema de archivos más utilizado en Linux, un sistema que es la cuarta evolución de la familia extended filesystems y que es ampliamente utilizando en dispositivos con kernel Linux, no solo en dispositivos Android.

Por otro lado, tenemos F2FS, Flash-Friendly File System, un sistema de archivos creado por Kim Jaegeuk en Samsung y que se integró en el kernel de Linux en la versión 3.6, en octubre de 2012. Como su propio nombre indica, este sistema está diseñado y optimizado para funcionar en memorias flash como las que equipan nuestros smartphones o tablets.

SEGURIDAD EN ANDROID

Arquitectura de seguridad en Android

Android es una plataforma móvil moderna que fue diseñada para ser de carácter libre. Las aplicaciones de Android hacen uso de hardware y software avanzados, así como de datos tanto locales como servidos, expuestos a través de la plataforma para ofrecer innovación y calidad a los consumidores. Para proteger esa calidad, la plataforma ofrece un entorno de aplicación que garantice la seguridad de los usuarios, datos, aplicaciones, el dispositivo, y la red.

Proporcionar una buena seguridad a una plataforma de código abierto como Android, requiere de una robusta arquitectura de seguridad y de rigurosos programas de seguridad. Android fue diseñado con varias capas de seguridad que proporcionan la flexibilidad necesaria para una plataforma de carácter abierto como esta, mientras que proporcionan la protección para todos los usuarios de la plataforma.

Los controles de seguridad fueron diseñados para reducir la carga de la misma sobre los desarrolladores. La seguridad inteligente hace que los desarrolladores puedan trabajar y confiar en los flexibles controles de seguridad proporcionados en la plataforma. Así mismo, los desarrolladores menos familiarizados con la seguridad estarán protegidos de la falta de esta última.

Android también fue diseñado teniendo en mente a los usuarios de los dispositivos bajo la plataforma. Dichos usuarios disponen de visibilidad en todo momento de cómo funcionan las aplicaciones y el control que tienen sobre las mismas.

A continuación en este proyecto vamos a describir los objetivos del programa de seguridad de Android y los fundamentos de la arquitectura de seguridad que emplea.

Android pretende ser el sistema operativo más seguro y útil para plataformas móviles por proponer controles de seguridad de sistemas operativos tradicionales para:

- Proteger los datos del usuario.
- Proteger los recursos del sistema (incluyendo la conexión de red).
- Proporcionar el aislamiento de aplicaciones.

Para lograr estos objetivos, Android proporciona las siguientes características clave de seguridad:

- Una seguridad robusta a nivel de sistema operativo a través del núcleo de Linux.
- "Sandbox" de aplicación obligatorio para todas las aplicaciones.
- Seguridad en la comunicación entre procesos.
- Firma de aplicaciones.
- Permisos de aplicación y usuarios.

Seguridad a nivel de sistema operativo

A nivel de sistema operativo, la plataforma Android ofrece la seguridad del kernel Linux, así como una comunicación segura entre procesos (IPC) permitiendo la posibilidad de comunicación segura entre aplicaciones que se ejecutan en diferentes procesos. Estas características de seguridad a nivel de sistema operativo aseguran que incluso el código nativo del sistema se ve limitado por el recinto de seguridad de aplicaciones. Ya sea este código el resultado del comportamiento de las aplicaciones ya incluidas, o la explotación de una vulnerabilidad de la aplicación, el sistema impedirá a la aplicación atacante hacer daño a otras aplicaciones, el sistema Android, o el propio dispositivo.

Seguridad a nivel de núcleo

La base de la plataforma Android es el núcleo de Linux. El kernel de Linux ha sido de uso generalizado durante años, y se utiliza en millones de entornos de seguridad. A lo largo de su historia ha sido constantemente revisado, atacado, y mejorado por miles de desarrolladores, Linux se ha convertido en un núcleo estable y seguro ganándose la confianza de muchas empresas y profesionales de la seguridad.

Como base para un entorno de computación móvil, el kernel de Linux ofrece a Android varias características de seguridad clave, incluyendo:

- Un modelo basado en permisos de usuario.
- Aislamiento de procesos.
- Mecanismo extensible para seguridad IPC.
- La capacidad de eliminar partes innecesarias y potencialmente inseguras del kernel.

Como sistema operativo multiusuario, un objetivo fundamental de seguridad del kernel de Linux es aislar los recursos de un usuario de otros. La filosofía de seguridad de Linux es la protección de recursos de los usuarios entre sí.

Seguridad a nivel de capa de aplicaciones

La plataforma Android aprovecha las ventajas de protección de usuarios que proporciona Linux, basándose en dicha protección como medio para identificar y aislar los recursos de las aplicaciones. El sistema Android asigna un identificador único de usuario (UID) a cada aplicación de Android y las ejecuta con ese identificador de usuario en procesos separados. Este enfoque es diferente de otros sistemas operativos (incluyendo la

tradicional configuración de Linux), donde múltiples aplicaciones se ejecutan con los permisos del mismo usuario.

Este mecanismo establece un recinto de seguridad para las aplicaciones a nivel de kernel. El núcleo refuerza la seguridad entre las aplicaciones y el sistema operativo a nivel de proceso, a través de los estándares que proporciona Linux, son el identificador de usuario e identificador de grupo que se asignan a las aplicaciones.

De forma predeterminada, las aplicaciones no pueden interactuar entre sí y tienen un acceso limitado al sistema operativo. l recinto de seguridad es simple, fácil de auditar, y está basado en décadas de experiencia en separación de procesos y permisos de archivos mediante el sistema de usuarios de estilo UNIX.

Desde que el mecanismo de recinto de seguridad de aplicación se encuentra en el núcleo, este modelo de seguridad se extiende al código nativo y aplicaciones integradas del sistema operativo. Todo el software por encima del núcleo se ejecutan dentro del recinto de seguridad de aplicaciones. En algunas plataformas, los desarrolladores se ven obligados a un marco de desarrollo en específico, un conjunto de APIs, o un lenguaje con el fin de reforzar la seguridad. En Android, no hay restricciones de cómo una aplicación debe ser escrita o desarrollada para reforzar la seguridad, en este sentido, el código nativo es tan seguro como código interpretado en Android.

En Android todas las aplicaciones y los recursos se encuentran en el recinto a nivel del sistema operativo. Un error de corrupción de memoria sólo permitirá la ejecución de código arbitrario en el contexto de esa aplicación en particular, con los permisos establecidos por el sistema operativo y no afectará a otras.

Al igual que todos los elementos de seguridad, el recinto de seguridad de aplicaciones no es irrompible. Sin embargo, al salir del recinto de seguridad de aplicaciones en un dispositivo configurado correctamente, se puede poner en peligro la seguridad del kernel de Linux.

Partición del sistema y modo seguro

La partición del sistema es aquella que contiene el kernel de Linux, las librerías, el tiempo de ejecución, el framework de aplicaciones y las aplicaciones. Esta partición es de sólo lectura. Cuando un usuario inicia el equipo en modo seguro, sólo se inicia la partición del sistema, por lo que sólo las aplicaciones básicas de Android estarán disponibles, lo que asegura que el usuario puede iniciar su teléfono con un ambiente libre de aplicaciones de terceros y por tanto en un modo seguro.

Permisos del sistema de archivos

Los permisos en el sistema de archivos Linux aseguran que un usuario no puede leer ni escribir archivos de otro usuario. Dado que en Android las aplicaciones se ejecutan como su propio usuario, una aplicación no puede leer ni escribir archivos creados por otra aplicación, a menos que el desarrollador de la misma los exponga de manera explícita.

Cifrado del sistema de archivos

Desde la versión 3.0 de Android se proporciona cifrado completo de los ficheros. Todos los datos se cifran en el kernel utilizando una clave derivada de la contraseña del usuario y para evitar ataques de fuerza bruta, ésta

se combina con una sal al azar y un hash al momento de la encripción. Para proporcionar resistencia contra ataques de diccionario, Android cuenta con reglas de complejidad de contraseñas que puede configurar el administrador del dispositivo y son ejecutadas por el sistema operativo.

El cifrado de disco en Android se basa en "dm-crypt" (provee cifrado transparente de dispositivos de bloque utilizando la nueva cryptoapi de Linux 2.6), que es una característica del núcleo que funciona en la capa de bloque del dispositivo. Por lo tanto, no se puede utilizar con "YAFFS", ya que esto trata directamente con chips flash "NAND", pero sí funciona con "emmc" y dispositivos flash similares que se presentan en el núcleo como dispositivos de bloque. El sistema de archivos actual preferido para usar en estos dispositivos es "ext4", sin embargo, esto es independiente de si se utiliza el cifrado o no.

Protección por contraseña

Android se puede configurar para validar contraseñas introducidas por el usuario antes de proporcionar el acceso de este al dispositivo. Además de prevenir el uso no autorizado del dispositivo, la contraseña protege la clave para el cifrado de sistema de ficheros completo.

El uso de una contraseña y reglas de complejidad de contraseñas puede ser requerido por el administrador del dispositivo.

Seguridad en la gestión de memoria

Android incluye muchas características que hacen que los problemas más comunes de seguridad resulten más difíciles de explotar. El SDK de Android, los compiladores y el sistema operativo usan herramientas que hacen más difícil explotar los problemas de corrupción de memoria, incluyendo:

- Disposición aleatoria del espacio de direcciones (ASLR) ubica al azar las claves en memoria.
- Hardware basado en la tecnología "No eXecute" (NX) para evitar la ejecución de código en la pila de memoria.
- "ProPolice" para evitar el desbordamiento del buffer de la pila.
- "safe_iop" para reducir el desbordamientos de enteros.
- Ampliaciones de "OpenBSD dlmalloc" para evitar vulnerabilidades "doble free ()" y para prevenir los ataques al fragmento de consolidación. Estos ataques son una forma común de explotar daños en la pila.
- "OpenBSD calloc" para evitar el desbordamientos de enteros durante la asignación de memoria.
- "Linux mmap_min_addr ()" para mitigar la referente escalada de privilegios de puntero nulo.

Permisos de "root" en los dispositivos

En Android sólo el kernel y un subconjunto pequeño de aplicaciones principales se ejecutan con permisos de root. Los permisos de root pueden modificar el sistema operativo, el núcleo y cualquier otra aplicación, además tiene acceso completo a los datos de aplicación. Si un usuario cambia los permisos del dispositivo y otorga permisos de root a las aplicaciones, está aumentando el riesgo de seguridad de aplicaciones maliciosas.

Android ha permitido que los permisos de root se puedan configurar ya que esta es una propiedad importante para los desarrollares y para aquel- los usuarios que por ejemplo quieren permitir la instalación de un sistema operativo alternativo.

Seguridad en las aplicaciones Android

APIs protegidas y firma de aplicaciones

De manera predeterminada, una aplicación Android sólo puede acceder a una gama limitada de los recursos del sistema. El sistema gestiona el acceso a los recursos de las aplicaciones Android que, si se usan incorrectamente o maliciosamente, pueden afectar desfavorablemente a la experiencia del usuario, la red, o los datos en el dispositivo.

Estas restricciones se aplican en una variedad de formas diferentes. Algunas capacidades se ven limitadas por una falta intencional de la API a la funcionalidad sensible (por ejemplo, no hay API de Android para la manipulación directa de la tarjeta SIM). En algunos casos, la separación de roles proporciona una medida de seguridad, al igual que con el aislamiento por la aplicación de almacenamiento. En otros casos, las API son sensibles para uso de aplicaciones de confianza y protegido a través de un mecanismo de seguridad conocido como permisos.

Para hacer uso de la API protegida en el dispositivo una aplicación debe definir las capacidades o funcionalidades que necesita en su archivo “manifest.xml”. En los preparativos para instalar una aplicación, el sistema muestra una interfaz al usuario que indica los permisos solicitados por la aplicación que se va a proceder a instalar y le pregunta si desea continuar con la instalación. Si el usuario continúa con la instalación, el sistema acepta que el usuario ha concedido todos los permisos solicitados. El usuario no puede conceder o denegar permisos individuales, este debe conceder o denegar todos los permisos solicitados en bloque.

Una vez concedida dicha solicitud al usuario se aplican los permisos a la aplicación a medida que esta es instalada. Para evitar la confusión de los usuarios, el sistema no avisa al usuario de nuevo de los permisos concedidos a la aplicación. Las aplicaciones que se incluyen en el núcleo del sistema operativo o por un paquete de OEM (fabricante de equipamiento original) no solicitan los permisos al usuario. Dichos permisos se eliminan si se desinstala la aplicación, por lo que un nueva instalación de la misma aplicación vuelve a mostrar la pantalla de los permisos al usuario.

Dentro de la configuración del dispositivo, los usuarios pueden ver los permisos para las aplicaciones que se han instalado previamente. Los usuarios también pueden desactivar algunas funciones a nivel global cuando lo requieren, por ejemplo, deshabilitar GPS, radio, o Wi-Fi.

En el caso de que una aplicación intentara utilizar una función o característica protegida que no se ha declarado en el archivo “manifest.xml” de la aplicación, la falta de autorización dará lugar a que una excepción de seguridad sea lanzada a la aplicación. Las comprobaciones de permisos API protegidos se aplican en el nivel más bajo posible para evitar la alusión de los mismos. Un ejemplo de la interfaz de usuario cuando se instala una aplicación al tiempo que se le solicita al mismo el acceso a la API protegida se muestra en la Figura 11 que mostraremos en el siguiente punto de este mismo capítulo.

Los permisos por defecto del sistema se describen en puntos posteriores de este mismo capítulo. Las aplicaciones pueden declarar sus propios permisos para usar otras aplicaciones, estos permisos no figuran en la ubicación anterior.

Cuando se define un permiso con el atributo “ProtectionLevel” le indicamos al sistema cómo el usuario debe ser informado de las aplicaciones que requieren de este permiso, o a cuales se le permite mantener este permiso. Detalles sobre la creación y el uso de los permisos de aplicaciones específicas se describen en puntos posteriores de este mismo capítulo.

Hay algunas funcionalidades o capacidades del dispositivo, tales como la función de enviar SMS, que no están disponibles para las aplicaciones externas o de terceros, pero que pueden ser utilizados por las aplicaciones pre-instaladas por el OEM (fabricante de equipamiento original). Estos permisos utilizan la autorización “signatureOrSystem”.

La solicitud de firma del código permite a los desarrolladores identificar el autor y responsable de la aplicación, las aplicaciones que intentan instalarse sin ser firmadas son rechazadas. Esta firma se hace por medio de certificados que son verificados en el momento de la instalación.

El firmado de código permite a los desarrolladores identificar al autor de la aplicación y actualizar su aplicación sin la creación de interfaces complicadas y permisos. Cada aplicación que se ejecuta en la plataforma Android debe ser firmada por el desarrollador. Las aplicaciones que intenta instalar sin la firma serán rechazadas tanto por Google Play como por el instalador de paquetes del dispositivo bajo la plataforma Android.

En Google Play, la firma de la aplicación proporciona la confianza de Google para con los desarrolladores y la confianza de estos con su aplicación. Los desarrolladores saben que su aplicación es proporcionada, sin modificaciones a los dispositivos Android y estos son los únicos responsables del comportamiento de la aplicación.

En Android, la firma de la aplicación es el primer paso para emplazar la misma en el recinto de seguridad de aplicaciones (“Application Sandbox”). El certificado de la aplicación firmada define qué ID de usuario está asociada a qué aplicación, ya que las diferentes aplicaciones se ejecutan bajo ID de usuario diferentes. La firma de la aplicación asegura que esta no puede acceder a cualquier otra aplicación, excepto a través de los bien definidos permisos IPC.

Cuando una aplicación (archivo “.apk”) se instala en un dispositivo Android, el gestor de paquetes comprueba que el archivo “.apk” ha sido debidamente firmado con el certificado incluido en mismo archivo. Si el certificado (o, más exactamente, la clave pública del certificado) coincide con la clave utilizada para firmar otros archivos “.apk” en el dispositivo, el nuevo archivo tiene la opción de especificar en su “manifest.xml” que compartirá su ID de usuario con los otros archivos firmados “.apk” de manera similar.

Las aplicaciones pueden ser firmadas por un tercero (OEM, operadores, el mercado alternativo) o auto-firmadas.Android ofrece firmado del código usando certificados auto-firmados que los desarrolladores pueden generar sin ayuda o permiso externo. Las aplicaciones no tienen que ser firmadas por una autoridad central, actualmente no realiza la verificación de certificados de aplicación CA (Autoridad de certificación).

Comunicación entre procesos

Los procesos pueden comunicarse con cualquiera de los mecanismos tradicionales de tipo UNIX. Los ejemplos incluyen por ejemplo el sistema de archivos, sockets locales o señales. Sin embargo, los permisos Linux siguen siendo válidos.

Android también proporciona nuevos mecanismos de IPC:

- Binder: una ligera funcionalidad basada en el mecanismo de llamada a procedimiento remoto diseñado para un alto rendimiento cuando se realiza durante el mismo proceso y/o las llamadas entre procesos."Binder" es implementado utilizando un controlador de Linux personalizado.
- Services: pueden proporcionar interfaces de acceso directo usando "Binder".
- Intents: Un "Intent" es un simple objeto de mensaje que representa una "intención" de hacer algo. Por ejemplo, si la aplicación desea mostrar una página web, expresa su "intención" para ver la dirección URL mediante la creación de una instancia "Intent" y la entrega fuera del sistema. El sistema detecta alguna otra pieza de código (en este caso, el navegador) que sabe cómo manejar esa intención, y lo ejecuta. Las intenciones también se puede utilizar para transmitir eventos interesantes (como la notificación) para todo el sistema.
- ContentProviders: es un almacén de datos que proporciona acceso a los datos en el dispositivo, el ejemplo clásico de "ContentProvider" es el que se utiliza para acceder a la lista de los contactos del usuario. Una aplicación puede acceder a los datos que otras aplicaciones han expuesto a través de un "ContentProvider", al igual que una aplicación también puede definir sus propios "ContentProviders" para exponer sus propios datos.

Acceso a la tarjeta SIM

El acceso de bajo nivel a la tarjeta SIM no está disponible para aplicaciones externas o de terceros. El sistema operativo gestiona todas las comunicaciones con la tarjeta SIM, incluyendo el acceso a la información personal (contactos) en la memoria de la tarjeta SIM. Las aplicaciones tampoco pueden acceder a los comandos AT (comandos UNIX que permiten programar la ejecución de uno o varios programas en un momento futuro), ya que estos son administrados exclusivamente por la "Radio Interface Layer" (RIL - es la capa del sistema operativo que proporciona una interfaz de radio para hardware y módem). El EIR (es una base de datos en la que existe información sobre el estado de los teléfonos móviles) no proporciona API de alto nivel para estos comandos.

Acceso a los datos personales

Android ha puesto un API que proporciona acceso a los datos del usuario en el conjunto de API protegidos. Con el uso cotidiano, los dispositivos Android acumulan datos personales proporcionados por el usuario para las aplicaciones de terceros instaladas en el dispositivo por los mismos. Las aplicaciones que deciden compartir esta información pueden utilizar las comprobaciones de los permisos del sistema operativo Android proteger los datos de las aplicaciones externas o de terceros.

Los proveedores de contenido del sistema es probable que contengan información personal o de identificación personal, como contactos y/o calendarios que han sido creados con permisos claramente identificados. Esta granularidad proporciona al usuario una clara indicación de los tipos de información que puede ser proporcionada a la aplicación. Durante la instalación, una aplicación ajena o de terceros puede solicitar permiso o varios permisos para acceder a estos recursos. Si le es otorgado dicho permiso o conjunto de permisos, la aplicación podrá ser instalada y tendrá acceso a los datos solicitados en cualquier momento cuando se instale.

Todas las aplicaciones que recopilan información personal tendrán, por defecto, acceso limitado a los datos, únicamente por dichas aplicaciones en específico. Si una aplicación opta por tener sus datos disponibles para otras aplicaciones a través del IPC, la aplicación que concede el acceso a su información puede aplicar los permisos del mecanismo IPC, que son impuestos por el sistema operativo para el intercambio de información entre procesos.

Entrada de datos por dispositivos

Los dispositivos bajo la plataforma Android a menudo proporcionan la entrada de datos sensibles o de alto nivel que permiten a las aplicaciones interactuar con el entorno circundante, tales son como la cámara, el micrófono o el GPS. Para acceder una aplicación externa o de terceros a los dispositivos de este tipo, primero se le debe proporcionar explícitamente el acceso por parte del usuario mediante el uso de los permisos del sistema operativo Android. Tras la instalación, el instalador le pedirá al usuario la solicitud del permiso para el sensor por su nombre.

Si una aplicación quiere saber la ubicación del usuario, la aplicación requiere un permiso para acceder a la ubicación del usuario. Tras la instalación, el instalador le preguntará al usuario si la aplicación puede acceder a la ubicación del usuario. En cualquier momento, si el usuario no quiere que cualquier aplicación pueda acceder a su ubicación, el usuario puede ejecutar la “Configuración” de la aplicación, ir al apartado “Ubicación y seguridad”, y desactive la casilla “Utilizar redes inalámbricas” y “Habilitar satélites GPS”. Esto desactivará los servicios de localización para todas las aplicaciones en el dispositivo del usuario.

Actualizaciones del sistema operativo Android

Android proporciona las actualizaciones del sistema, tanto para propósitos de seguridad como para otras funciones relacionadas.

Hay dos formas de actualizar el código en la mayoría de dispositivos Android: “on-the-air” (actualizaciones OTA) o actualizaciones cargadas. Las actualizaciones OTA se pueden lanza a lo largo de un período de tiempo definido o ser lanzadas a todos los dispositivos a la vez, dependiendo de cómo el OEM y/o distribuidor quieran lanzar las actualizaciones.Las actualizaciones cargadas se pueden proporcionar desde una ubicación central, los usuarios descargan un archivo zip a su equipo o directamente a su dispositivo Android. Una vez que la actualización se ha copiado o descargado a la tarjeta SD del dispositivo, Android reconocerá el archivo de actualización, comprobará su integridad y autenticidad, y actualizará automáticamente el dispositivo.

El NDA tiene la obligación de asegurar que el problema de seguridad no se haga público antes de disponer de una solución y, por lo tanto, poner a los usuarios en una situación de riesgo. Muchos miembros de la OHA (alianza comercial de 78 compañías para desarrollar estándares abiertos para dispositivos móviles) ejecutan su propio código en los dispositivos Android, como el gestor de arranque, los controladores de Wi-Fi, y la radio. Una vez que el equipo de seguridad de Android es notificado de un problema de seguridad en este código promocional, se consultará a los miembros de la OHA para encontrar rápidamente una solución al problema en cuestión y/o problemas similares. Sin embargo, el miembro de la OHA, que desarrolló el código defectuoso es responsable en última instancia de solucionar el problema.

Si de una vulnerabilidad peligrosa no se conoce al responsable (por ejemplo, si es enviada a un foro público sin previo aviso), Google y/o el Proyecto de Código Abierto Android trabajarán lo más rápido posible para crear un parche o solución al problema. El parche será lanzado al público (y socios) cuando sea probado y esté listo para su uso.

ID de usuario y acceso

Android otorga a cada paquete un identificador de usuario Linux (UID). Este identificador se mantiene constante durante la vida del paquete en ese dispositivo. En un dispositivo diferente, el mismo paquete puede tener un UID diferente, lo importante es que cada paquete tiene un UID distinto en un dispositivo determinado.

Debido al refuerzo de seguridad que se aplica a nivel de proceso, el código de dos paquetes cualesquiera normalmente no puede ejecutarse en el mismo proceso, ya que necesitan para funcionar diferentes identificadores de usuarios Linux. Podemos utilizar el atributo “sharedUserId” en el manifiesto de etiquetas “AndroidManifest.xml” de cada paquete que les asigna el mismo ID de usuario. De esta manera, por razones de seguridad los dos paquetes son tratados como la misma aplicación, con el mismo ID de usuario y permisos de archivos. Tenga en cuenta que el fin de mantener la seguridad, sólo dos aplicaciones firmadas con la misma firma (y su interés en la misma sharedUserId) tendrá la misma identificación de usuario.

ESCANEO DE DISPOSITIVOS MÓVILES IOS

Estudio de las herramientas a utilizar

A continuación nombraremos el equipamiento necesario y algunas de las herramientas que utilizaremos en esta primera fase de escaneo y una breve descripción sobre ellas:

Equipo técnico:

- Ordenador(PC, Mac o Linux).
- Conexión a Internet.
- Cables de datos de enlace para dispositivos IOS.

Herramientas necesarias:

- Nmap: Es una herramienta gratuita de escaneo de redes que permite identificar qué servicios se están ejecutando en un dispositivo remoto, así como la identificación de estos, y existencia de filtros o firewalls, entre otros.
- Wireshark: Wireshark es una herramienta multiplataforma open-source de análisis de red, producto de la evolución de Ethereal. Se utiliza para realizar análisis y solucionar problemas en redes de comunicaciones, efectuar auditorías de seguridad, para desarrollo de software y protocolos, y como una herramienta didáctica. Cuenta con todas las características estándar de un analizador de protocolos.
- Santoku Linux: Es una distribución de Linux basada en Lubuntu 14.04 especialmente diseñada y complementada con una serie de herramientas para llevar cabo tareas de Análisis de Malware, análisis forense de smartphones, y pruebas de seguridad de Aplicaciones. Posee una gran cantidad de herramientas ya preinstaladas y preconfiguradas para ayudarnos en futuras verificaciones y pruebas móviles. Entre ellas podemos destacar ExifTool, Iphone Backup Analyzer y SSL strip.
- Kali Linux: Es una distribución basada en Debian GNU/Linux diseñada principalmente para la auditoría y seguridad informática en general. Una de las principales virtudes de Kali Linux son las más de 300 herramientas y aplicaciones relacionadas con la seguridad informática que incluye esta distribución. Algunas aplicaciones mencionadas anteriormente las podemos encontrar en esta distro.

Exploración del dispositivo

En primer lugar al observar el dispositivo móvil lo primero que vamos a realizar será el reconocimiento del dispositivo de forma visual. Este punto es importante porque nos ahorrará tiempo a la hora de decidir que métodos o herramientas utilizar en las fases posteriores. Por tanto debemos fijarnos principalmente en el diseño del dispositivo para poder identificar el modelo exacto, y para ello tendremos que prestar mayor atención al tamaño, acabados, color y forma. En este caso, como vamos a analizar los dispositivos que utilizan el sistema operativo iOS, solo tendremos que centrarnos en los modelos del fabricante Apple Inc.

Para poder reconocer e identificar el modelo exacto podemos utilizar 2 métodos más precisos. El primero es mediante la comprobación del código de modelo. Para ello exploraremos la cubierta posterior, donde en un

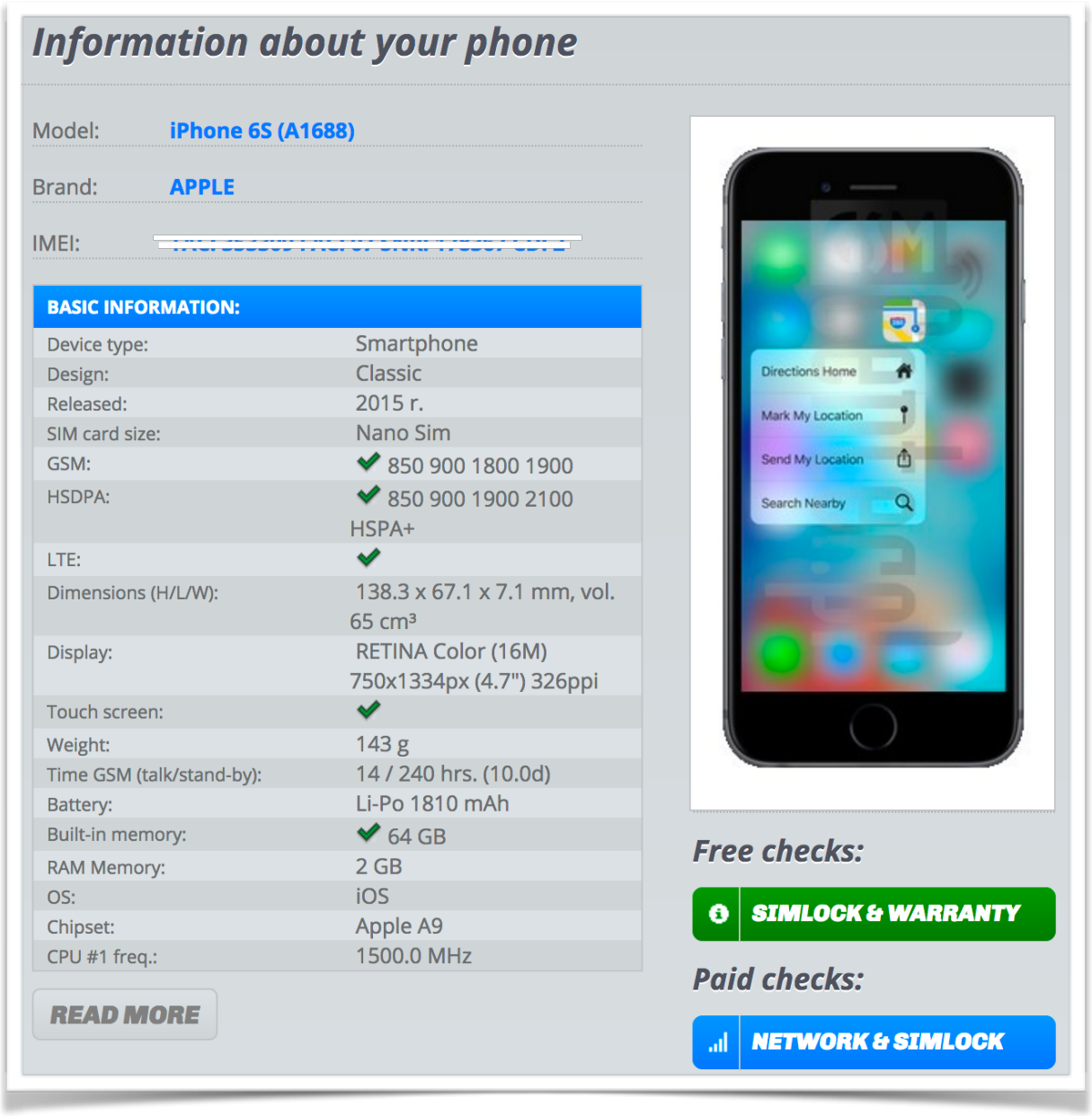
Captura de la web de Apple, para identificar el modelo de iPhone. Fuente: www.apple.com

pequeño rótulo aparece el número de modelo. Con este número identificaremos el dispositivo en la siguiente web del fabricante https://support.apple.com/es-es/HT201296.

En la anterior captura de esta web podemos observar las características y el número de modelo de un dispositivo Iphone 7.

El segundo método que podemos hacer servir será a través del rótulo de la bandeja SIM. En esta bandeja aparece rotulado el IMEI y el serial del dispositivo, y con el número de IMEI podremos obtener la información relacionada con nuestro dispositivo, haciendo la comprobación en http://www.imei.info.

Si tenemos el dispositivo conectado a nuestra red Wifi o tenemos la posibilidad de conectarlo, podremos lanzar un escaneo de puertos sobre el propio terminal.



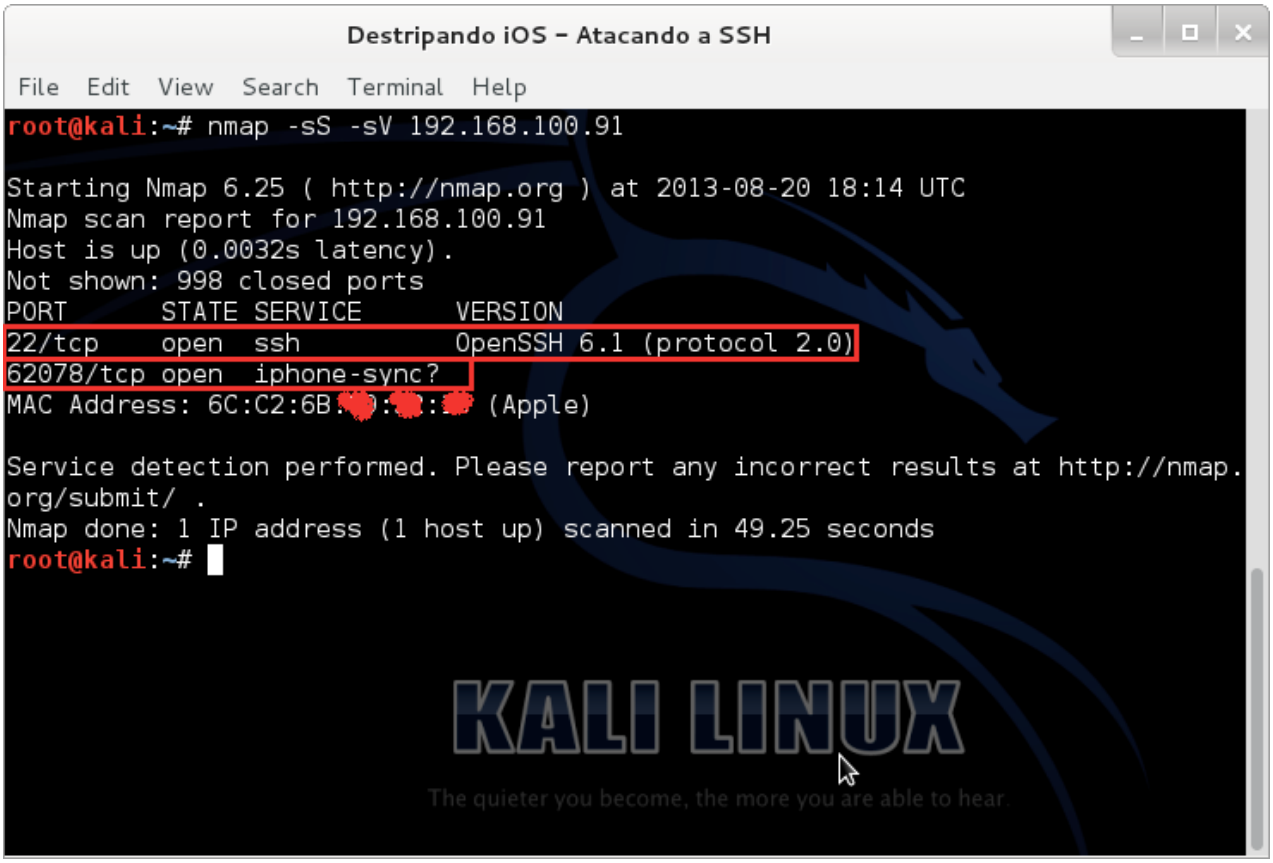
Captura de la web donde identificar el modelo de iPhone. Fuente: www.imei.info

Para ello necesitaremos conocer la ip que el dispositivo móvil tiene asignada dentro de nuestra red o simplemente realizando un escaneo para localizar los equipos conectados dentro de la misma. Para realizar esta tarea haremos uso de la herramienta Nmap.

Nmap está disponible para diferentes Sistemas Operativos como: MS Windows, GNU/LiNUX, Mac OS X, y otras plataformas UNIX como son Solaris, FreeBSD, NetBSD y OpenBSD, etc. Se puede obtener en la siguiente dirección web: http://nmap.org/download.html

Con el siguiente comando haremos la comprobación:

Nmap –sS –sV 192.168.100.91



Captura del resultado del escaneo.

Los dispositivos iOS en estado normal (sin Jailbreak) tan solo tienen un puerto TCP/IP en modo Listening (a la escucha), ese puerto es el 62078/TCP, que es el que utiliza iTunes para sincronizarse con los dispositivos iOS.

Sabiendo esto, podemos intuir que si un host tiene este puerto en modo Listening con muchas posibilidades se tratara de un dispositivo iOS.

La herramienta Nmap es capaz de identificar un dispositivo iOS si tiene este puerto abierto, debido a que Nmap necesita tener al menos un puerto abierto para poder realizar todos los test de detección.

Si a un dispositivo iOS se le ha realizado un Jailbreak, tendrá a mayores con muchas posibilidades el puerto 22 TCP/IP. Se trata del servicio utilizado por el paquete OpenSSH que permite la gestión remota del dispositivo a través del protocolo SSH (Secure SHell).

Análisis de la información obtenida

En esta fase, una vez tenemos claro cuál es el modelo del dispositivo, lo siguiente que tendremos que hacer es obtener la versión de iOS. Para ello habrá que ver que versiones de iOS han sido compatibles con ese

iOS 8.x:

	iOS 8.0	iOS 8.0.1	iOS 8.0.2	iOS 8.1	iOS 8.1.1	iOS 8.1.2	iOS 8.1.3	iOS 8.2	iOS 8.3	iOS 8.4	iOS 8.4.1
iPhone 4S	Pangu8, TaiG, PP	Pangu8, TaiG, PP	Pangu8, TaiG, PP	Pangu8, TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	No JB
iPhone 5	Pangu8, TaiG, PP	Pangu8, TaiG, PP	Pangu8, TaiG, PP	Pangu8, TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	No JB
iPhone 5c	Pangu8, TaiG, PP	Pangu8, TaiG, PP	Pangu8, TaiG, PP	Pangu8, TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	No JB
iPhone 5s	Pangu8, TaiG, PP	Pangu8, TaiG, PP	Pangu8, TaiG, PP	Pangu8, TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	No JB
iPhone 6	Pangu8, TaiG, PP	Pangu8, TaiG, PP	Pangu8, TaiG, PP	Pangu8, TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	No JB
iPhone 6 Plus	Pangu8, TaiG, PP	Pangu8, TaiG, PP	Pangu8, TaiG, PP	Pangu8, TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	No JB
iPod Touch 5	Pangu8, TaiG, PP	Pangu8, TaiG, PP	Pangu8, TaiG, PP	Pangu8, TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	No JB
iPod Touch 6	-	-	-	-	-	-	-	-	-	TaiG, PP	No JB
iPad 2	Pangu8, TaiG, PP	Pangu8, TaiG, PP	Pangu8, TaiG, PP	Pangu8, TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	No JB
iPad 3	Pangu8, TaiG, PP	Pangu8, TaiG, PP	Pangu8, TaiG, PP	Pangu8, TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	No JB
iPad 4	Pangu8, TaiG, PP	Pangu8, TaiG, PP	Pangu8, TaiG, PP	Pangu8, TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	No JB
iPad Air	Pangu8, TaiG, PP	Pangu8, TaiG, PP	Pangu8, TaiG, PP	Pangu8, TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	No JB
iPad Air 2	-	-	-	Pangu8, TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	No JB
iPad Mini 1	Pangu8, TaiG, PP	Pangu8, TaiG, PP	Pangu8, TaiG, PP	Pangu8, TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	No JB
iPad Mini 2	Pangu8, TaiG, PP	Pangu8, TaiG, PP	Pangu8, TaiG, PP	Pangu8, TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	No JB
iPad Mini 3	-	-	-	Pangu8, TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	TaiG, PP	No JB

↑ Back to Top ↑

iOS 7.x:

	iOS 7.0	iOS 7.0.1	iOS 7.0.2	iOS 7.0.3	iOS 7.0.4	iOS 7.0.5	iOS 7.0.6	iOS 7.1	iOS 7.1.1	iOS 7.1.2
iPhone 4	evasi0n7	-	evasi0n7	evasi0n7	evasi0n7	-	evasi0n7	Pangu	Pangu	Pangu
iPhone 4S	evasi0n7	-	evasi0n7	evasi0n7	evasi0n7	-	evasi0n7	Pangu	Pangu	Pangu
iPhone 5	evasi0n7	-	evasi0n7	evasi0n7	evasi0n7	-	evasi0n7	Pangu	Pangu	Pangu
iPhone 5c	evasi0n7 (Modified)	evasi0n7	evasi0n7	evasi0n7	evasi0n7	evasi0n7	evasi0n7	Pangu	Pangu	Pangu
iPhone 5s	evasi0n7 (Modified)	evasi0n7	evasi0n7	evasi0n7	evasi0n7	evasi0n7	evasi0n7	Pangu	Pangu	Pangu
iPod Touch 5	evasi0n7	-	evasi0n7	evasi0n7	evasi0n7	-	evasi0n7	Pangu	Pangu	Pangu
iPad 2	evasi0n7	-	evasi0n7	evasi0n7	evasi0n7	-	evasi0n7	Pangu	Pangu	Pangu
iPad 3	evasi0n7	-	evasi0n7	evasi0n7	evasi0n7	-	evasi0n7	Pangu	Pangu	Pangu
iPad 4	evasi0n7	-	evasi0n7	evasi0n7	evasi0n7	-	evasi0n7	Pangu	Pangu	Pangu
iPad Air	-	-	-	evasi0n7	evasi0n7	-	evasi0n7	Pangu	Pangu	Pangu
iPad Mini 1	evasi0n7	-	evasi0n7	evasi0n7	evasi0n7	-	evasi0n7	Pangu	Pangu	Pangu
iPad Mini 2	-	-	-	evasi0n7	evasi0n7	-	evasi0n7	Pangu	Pangu	Pangu

Captura de la web donde aparecen algunas tablas con los modelos y versiones. Fuente: http://rcrepo.com/jailbreak/

terminal. Podremos utilizar la información que ofrecen las siguientes webs, que además nos servirán para comprobar, en las siguientes fases, si nuestro modelo y versión de móvil tiene la posibilidad de realizar el Jailbreak y con qué herramienta.

http://rcrepo.com/jailbreak/tables.html

https://www.reddit.com/r/jailbreak/wiki/escapeplan/guides/jailbreakcharts

Para identificar la versión de iOS que está instalada en el dispositivo, podemos diferenciar visualmente la versión existente, aunque será muy complicado conocer la versión exacta.

Si tenemos acceso al dispositivo porque conocemos el código de seguridad o este no está activado, podremos obtener la versión de iOS en la pantalla “Ajustes -> General -> Información”.

En el caso que no pudiéramos acceder al escritorio del terminal, un método que podemos intentar pero que requiere que el dispositivo esté conectado a una red Wifi conocida y que tengamos acceso, es esnifar el tráfico de red procedente del dispositivo móvil mediante alguna aplicación como Wireshark, e identificar mediante el valor User-Agent, la versión exacta de iOS que está utilizando.

Información del dispositivo accediendo desde “Ajustes”. Fuente: www.apple.com

Informe de resultados

Llegado a este punto, después de haber explorado el dispositivo y analizado la información obtenida, debemos tener claro qué modelo de smartphone estamos estudiando, su versión de sistema operativo iOS y en el mejor de los casos, si el dispositivo tiene el Jailbreak hecho. Aunque esto último lo trataremos en la siguiente fase del estudio.

EVALUACIÓN Y ACCESO AL SISTEMA EN IOS

Estudio de las herramientas a utilizar

Para continuar con la siguiente fase de auditorias y análisis de seguridad en smartphones vamos a necesitar algunas herramientas que nos ayudarán a realizar las pruebas correspondientes.

A continuación nombraremos el equipamiento necesario y algunas de las herramientas que utilizaremos en esta segunda fase de evaluación y acceso y una breve descripción sobre ellas:

Equipo técnico:

- Ordenador(PC, Mac o Linux).
- Conexión a Internet.
- Cables de datos de enlace para dispositivos IOS.

Herramientas necesarias:

- Tools para realizar Jailbreak: El jailbreak en IOS es equivalente al root de Android. El jailbreak permite saltarse algunas de las limitaciones de software impuestas por Apple en nuestros dispositivos iOS. Dependiendo de las versiones de IOS de nuestros dispositivos Apple, utilizaremos unos programas u otros. Más adelante comentaremos todas las herramientas existen para realizar este proceso.
- Cliente SSH: Nos servirá para conectarnos remotamente a los dispositivos una vez el root/jailbreak esté realizado. Permite ejecutar comandos y acceder a los datos de forma remota.
- Oxygen Forensic Suite: La herramienta permite realizar un análisis forense online, es decir, con el dispositivo encendido y conectado al equipo, y extraer una amplia cantidad de información de un smartphone.
- Santoku Linux: Es una distribución de Linux basada en Lubuntu 14.04 especialmente diseñada y complementada con una serie de herramientas para llevar cabo tareas de Análisis de Malware, análisis forense de smartphones, y pruebas de seguridad de Aplicaciones. Posee una gran cantidad de herramientas ya preinstaladas y preconfiguradas para ayudarnos en futuras verificaciones y pruebas móviles. Entre ellas podemos destacar ExifTool, Iphone Backup Analyzer y SSL strip.
- Kali Linux: Es una distribución basada en Debian GNU/Linux diseñada principalmente para la auditoría y seguridad informática en general. Una de las principales virtudes de Kali Linux son las más de 300 herramientas y aplicaciones relacionadas con la seguridad informática que incluye esta distribución. Algunas aplicaciones mencionadas anteriormente las podemos encontrar en esta distro.

Identificación de vulnerabilidades y proceso de Jailbreak

Proceso de Jailbreak

Para entender como se puede acceder a la totalidad de los datos de un smartphone, es necesario conocer el significado de Jailbreak, los tipos de Jailbreak existentes y cómo funcionan.

"Se denomina Jailbreak al proceso de suprimir algunas de las limitaciones impuestas por Apple en dispositivos que utilicen el sistema operativo iOS mediante el uso de kernels modificados. El jailbreak permite a los usuarios acceder por completo al sistema operativo, permitiendo al usuario descargar aplicaciones, extensiones y temas que no están disponibles a través de la App Store oficial. El jailbreak es una forma de escalado de privilegios." Fuente: es.wikipedia.org

Este proceso permite al usuario disponer de control completo sobre el dispositivo móvil y acceder al mismo como "root", o usuario privilegiado, y en concreto, al subsistema Unix subyacente en iOS (basado en Darwin BSD disponible en OS X), eliminando así los controles y/o restricciones establecidos por la plataforma iOS estándar y su fabricante Apple. Este proceso permite la instalación de aplicaciones, modificaciones y componentes del sistema no proporcionados a través de la tienda oficial de aplicaciones de Apple, denominada App Store. A diferencia de otras plataformas móviles como Android, en el caso de iOS, este proceso es necesario si el usuario desea ejecutar software no autorizado por Apple, aplicaciones de terceros no oficiales, es decir, no distribuidas a través de la App Store y, por tanto, código no firmado.

El proceso de jailbreak (que difiere en dispositivos móviles iOS con chip A4, como iPhone 4 e iPad, y dispositivos móviles basados en el chip A5, como iPhone 4S e iPad2, o A5X del iPad de 3a generación, o versiones posteriores del SoC, System-on-a-Chip, de Apple) permite, además de instalar aplicaciones de terceros, realizar modificaciones y añadir extensiones al sistema operativo y el sistema de ficheros de iOS, habitualmente a través del cliente de instalación de software Cydia. El proceso de jailbreak es utilizado por usuarios avanzados para acceder a estas funcionalidades de bajo nivel, pudiendo aplicar correcciones, modificar el interfaz de usuario con nuevas capacidades, y proporcionar acceso al sistema de ficheros y al interfaz de línea de comandos(shell).

El proceso de jailbreak se lleva a cabo a través de la explotación de vulnerabilidades en el hardware, iOS o alguna de las aplicaciones asociadas. En función del tipo de vulnerabilidad explotada el jailbreak permite modificar el entorno de usuario o incluso el proceso de arranque (boot) del dispositivo móvil.

Por otro lado, en la actualidad existen dos tipos de jailbreak en función de los requisitos necesarios para hacer uso del mismo: tethered o untethered. El primero de ellos, tethered, requiere conectar el dispositivo móvil a un ordenador durante su arranque, ya que el jailbreak hace uso de código disponible en el ordenador para permitir el arranque y evitar la detección de software no firmado en el terminal. El segundo, untethered, es independiente y no requiere conectar el dispositivo móvil a un ordenador durante el arranque, ya que el jailbreak modifica el código binario que comprueba y bloquea la existencia de código no firmado. Los dispositivos móviles iOS jailbroken ignoran el modelo de seguridad descrito e impuesto por Apple, ya que todas las aplicaciones que sean instaladas dispondrán de los máximos privilegios en el dispositivo("root"), exponiendo a los usuarios a código dañino que podría tomar control completo del terminal.

En el caso de un investigador o analista de seguridad, este puede intentar por diferentes medios la extracción de datos del dispositivo sin modificar el sistema original, sin embargo, en diferentes circunstancias será necesario como mejor alternativa explotar un Tethered Jailbreak que permita acceder a los datos y tras el reinicio volver al estado original. Esto conlleva, a documentar todo el proceso y los motivos por el cual se procede a realizar esta modificación en los sistemas iOS.

Las particularidades del proceso de actualizaciones en iOS, en algunos casos con relativamente largos periodos de tiempo hasta que una nueva actualización está disponible para los usuarios finales, es uno de los motivos que incentiva a algunos usuarios a realizar el proceso de jailbreak para así poder instalar la última versión de ciertos componentes o software en su dispositivo móvil.

Un proceso asociado al jailbreak, pero diferente, es el proceso de unlock o desbloqueo, que permite al usuario liberar el componente de radio o banda base (baseband) del iPhone o iPad para poder hacer uso del dispositivo móvil con cualquier operador de telefonía móvil, es decir, utilizando cualquier tarjeta SIM. Este proceso era necesario en las versiones iniciales de iPhone si se quería disponer de libertad para elegir el operador, ya que el modelo de negocio de Apple restringía el uso de iPhone a ciertos operadores de telefonía móvil. El modelo de negocio actual de Apple permite adquirir el iPhone e iPad libres, es decir, sin vinculación a un operador de telefonía móvil, no siendo necesario llevar a cabo el proceso de unlock por este motivo.

Herramientas de Jailbreak

Las herramientas disponibles para realizar Jailbreak dependen de las diferentes versiones de iOS. En este punto se van a recoger las versiones de cada una de ellas, para conocer por cual es posible decidirse en cada caso. También existe la posibilidad de elegir la versión correcta de forma interactiva, consultando en Internet las diferentes webs que ofrecen este servicio, como por ejemplo en http://es.jailbreakwizard.info.

La utilización de estos programas suele ser trivial, por lo que no se necesitan conocimientos técnicos avanzados para llevar a cabo el procedimiento.

Las herramientas existentes para dicho proceso son las siguientes:

- JailbreakMe: Se basa en la explotación de una vulnerabilidad mediante la visita con el dispositivo a una página web(http://www.jailbreakme.com). Esta ataca al sistema y realiza un Untethered Jailbreak. Es necesario que la sesión esté abierta, por lo tanto se requiere conocer el passcode del dispositivo previamente. Existen tres versiones:

    1) JailbreakMe: Para versiones de iOS 1.1.1 a 2.0.

    2) JailbreakMe 2.0: Para iOS 3.2.1 a 4.0.1.

    3) JailbreakMe 3.0: Para sistemas iOS 4.3 a 4.33 y Iphone CDMA en versiones 4.2.6 a 4.2.8.

- Purplera1n: Herramienta para realizar Jailbreak para el modelo de Iphone 3GS con iOS 3.0. Una característica notable de este software es que no requiere conocer el passcode o código de desbloqueo del dispositivo.

- Blackra1n: Esta es una evolución de la anterior, para Iphone 3G/3GS para las versiones de iOS 3.1, 3.1.1 y 3.1.2. No requiere passcode.
- Limera1n: Esta herramienta está desarrollada para realizar Tethered Jailbreak en dispositivos Iphone 3GS, Iphone 4 e iPad 1, es decir, todos los dispositivos con chipset A4. Utiliza un exploit de BootRom en el arranque y no requiere passcode.
- GreenPois0n: Esta herramienta utiliza el exploit SHATtier en el BootRom para saltarse la seguridad de Apple. Soporta los dispositivos iPad 1 con iOS 3.2.2, Iphone 3GS con iOS 4.1 e iPhone 4 con iOS 4.1 a 4.2.1. Realiza un Tethered Jailbreak y no es necesario conocer el código de desbloqueo del dispositivo.
- Redsn0w: Es una de las herramientas más conocidas y que más versiones existen. En sus primeras versiones se llamaba QuickPwn. Es el software por excelencia para los dispositivos Iphone 3GS, Iphone 4 e iPad 1. Las últimas versiones también son compatibles con Iphone 4S, iPad 2 e iPad 3. Esta herramienta permite hacer Tethered y Untethered Jailbreak de iOS 3 al 5.1.1. En iOS 6 solo soporta el Tethered. Existen más de 50 versiones de esta aplicación actualmente. Redsn0w se puede utilizar con dispositivos que tengan el chip A4 sin necesidad de conocer el passcode.
- Absinthe: Esta herramienta está desarrollada para realizar Jaibreak en dispositivos con chip A5, que son los iPhone 4S e iPad 2. Soporta las versiones de iOS 5 a 5.1.1. Para realizar el proceso de Jailbreak es necesario tener el dispositivo conectado a un equipo pareado o tener el passcode del terminal antes de hacer el Jailbreak para poder parearlo.
- Evasi0n: Herramienta para realizar Jailbreak en dispositivos con iOS 6 a 6.1.2. Es capaz de ejecutar código inicial en el dispositivo, utiliza las vulnerabilidades CVE-2013-0979 y CVE-2013-0981. También existe una versión conocida como Evasi0n 7, que sirve para dispositivos con versiones iOS 7.0 a 7.6 beta. Utiliza las vulnerabilidades CVE-2014-1273 y CVE-2014-1272.
- P0sixspwn: Este software permite realizar Untethered Jailbreak para las versiones de iOS 6.1.3 a 6.1.6 en dispositivos con chip A5x.
- TaiG Jailbreak: Taig lanzó en Junio del 2015 la herramienta de Jailbreak para iOS 8.1.3-8.3, apoya al untethered Jailbreak para 8.4, que a su vez esperó a que Apple lanzara la versión de iOS 8.4 para lanzar su versión de Jailbreak iOS 8.4. Para utilizar esta aplicación se necesita desactivar el passcode y el find My phone.
- Pangu: Herramienta para realizar Jailbreak en dispositivos con iOS 7.1 a 7.1.2, 8.0 a 8.1 y 9.0 a 9.3.3, aunque estas últimas versiones no lo permite en todos los dispositivos. Para realizar este Jailbreak se necesita tener el dispositivo en modo avión, desactivar find My phone y conectar el dispositivo al ordenador.
- PP: Aplicación para realizar Jailbreak en todas las versiones de iOS 8.X.

Como realizar el Jailbreak

Existen diferentes métodos para realizar el Jailbreak en un dispositivo iOS, dependiendo de la versión de iOS, del tipo de dispositivo y de la herramienta necesaria para realizar el proceso. Anteriormente hemos mencionado las diferentes herramientas existentes, y en este apartado explicaremos como realizarlo utilizando algunas de ellas.

El ejemplo de uso más sencillo es el caso de JailbreakMe, donde el proceso se ejecutaba simplemente accediendo a una página web.

A continuación hacemos una breve explicación de uso de otras herramientas muy populares como son RedSn0w y PP.

Utilización de RedSn0w en dispositivos con chipA4 e iOS 6

Lo primero será obtener el software necesario para realizar el proceso. Lo podremos descargar de su la página de Internet www.redsn0w.us.

1) El proceso es sencillo, conectamos el dispositivo mediante el cable de datos al equipo.

2) Apagamos el dispositivo y arrancamos la aplicación.

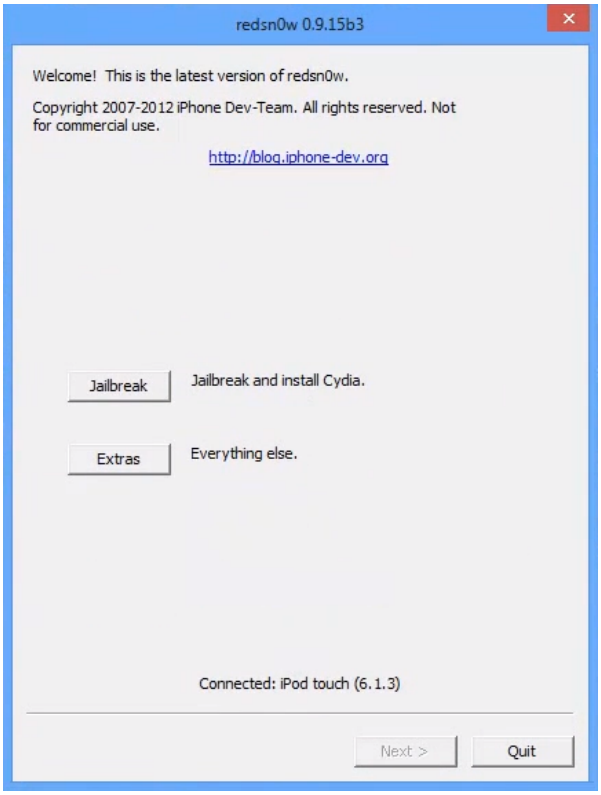
3) Hacemos clic en “Jailbreak” y seguimos el asistente.

4) Ponemos el dispositivo en modo DFU(device firmware update) , y automáticamente comenzará el proceso de realización del Jailbreak.

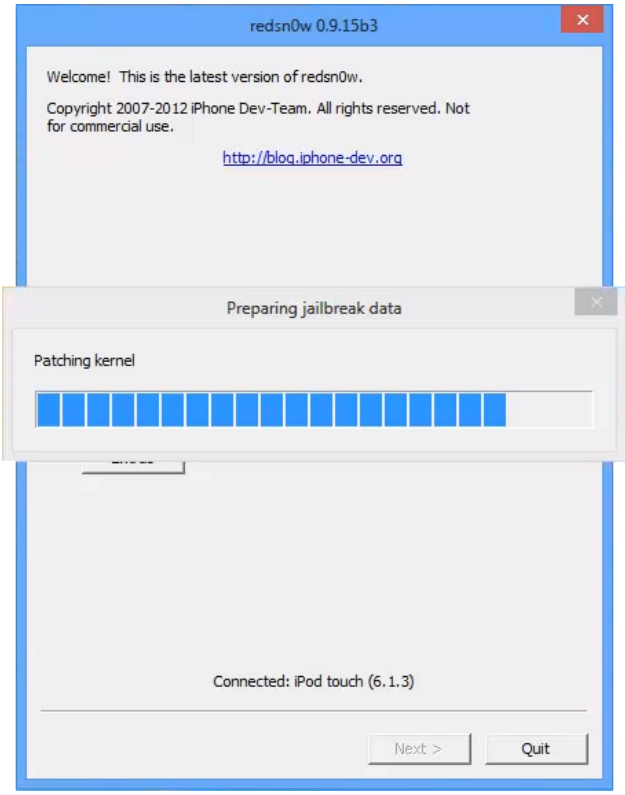
5) Se mostrará la información del exploit que se está lanzando, el parcheo del kernel y finalmente el esperado Jailbreak.

Suponiendo que no tenemos el passcode, se podría instalar Cydia* desde la propia herramienta ya que el dispositivo utiliza el chip A4.

* Cydia es la herramienta cliente que da acceso a la tienda de aplicaciones para sistemas iOS con Jailbreak.



Primeros pasos con la aplicación Redsn0w

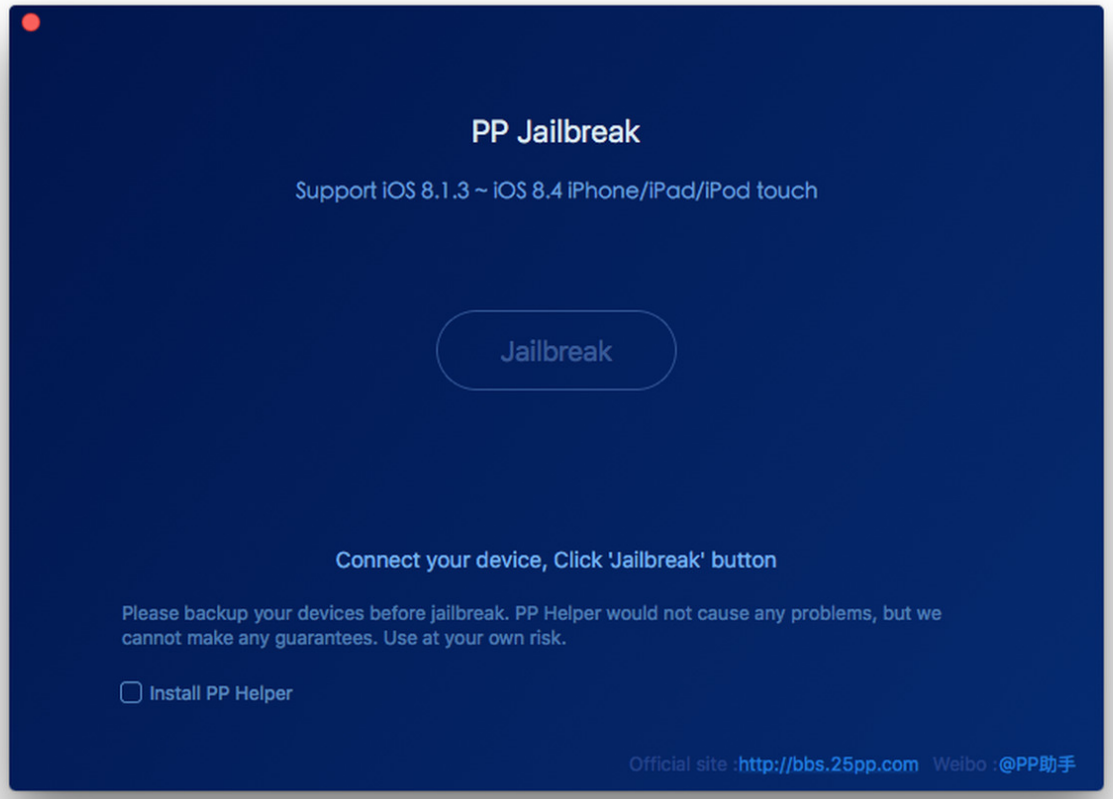


Momento de realización del Jailbreak

Utilización de PP para el Jailbreak en iOS 8

Este es el ejemplo de una de las herramientas más sencillas de utilizar. Esta aplicación permite realizar el Jailbreak en dispositivos Apple en cualquiera de las versiones de iOS 8.X.

El proceso es tan sencillo como conectar el dispositivo al equipo y pulsar sobre el botón Jailbreak para realizar el proceso de forma automática.



Pantalla de la aplicación para Jailbreak con un dispositivo con iOS 8.1.3

Vulnerabilidades públicas en iOS

Para poder conocer el nivel de seguridad y las posibles amenazas a las que está expuesto el dispositivo a analizar, debemos comprobar qué vulnerabilidades hay publicadas para la versión de iOS que tiene nuestro terminal. Para ello podremos consultar la web de CVE Details donde aparecen todas las vulnerabilidades reportadas desde los inicios de iOS, que hacen un total de 984. La web es la siguiente:

https://www.cvedetails.com/vulnerability-list/vendor_id-49/product_id-15556/Apple-Iphone-Os.html

También podemos consultar la página oficial de Apple siguiente, donde explica cada actualización y el problema que subsana de manera detallada.

https://support.apple.com/es-es/HT201222

Con el fin de proteger a sus clientes, Apple no revelará, discutirá ni confirmará problemas de seguridad hasta que se haya llevado a cabo una investigación y estén disponibles las revisiones o las versiones necesarias. Por lo tanto en esta web solo estarán las vulnerabilidades reconocidas y resueltas por el fabricante.

IOS 10.1

Disponible el 24 de octubre de 2016

CFNetwork Proxies

Disponible para: iPhone 5 y posterior, iPad 4.ª generación y posterior, iPod touch 6.ª generación y posterior.

Impacto: un atacante que se encontrase en una posición de red privilegiada podría filtrar información confidencial de los usuarios.

Descripción: existía un problema de suplantación de identidad en la gestión de las credenciales del proxy. Se ha solucionado el problema eliminando los mensajes emergentes de autenticación de la contraseña del proxy no solicitados.

CVE-2016-7579: Jerry Decime

Contactos

Disponible para: iPhone 5 y posterior, iPad 4.ª generación y posterior, iPod touch 6.ª generación y posterior.

Impacto: una aplicación podría mantener acceso a la Agenda tras revocarse el acceso en Ajustes.

Descripción: se solucionó un problema de control de acceso en la Agenda mejorando la validación del vínculo a archivo.

CVE-2016-4686: Razvan Deaconescu, Mihai Chiroiu (University POLITEHNICA de Bucarest); Luke Deshotels, William Enck (North Carolina State University); Lucas Vincenzo Davi, Ahmad-Reza Sadeghi (TU Darmstadt)

Vulnerabilidades solucionadas en la versión iOS 10.1. Fuente: www.apple.com

Dentro de cada versión aparecen detalladas las vulnerabilidades que han sido solucionadas. Indicando los modelos de dispositivos a los que afecta, el impacto, la descripción y el código de CVE.

En la imagen superior podemos observar lo que hemos comentado anteriormente. Donde podemos observar una parte de las vulnerabilidades solucionadas en la versión 10.1 de iOS.

Métodos de explotación y acceso

Acceso al dispositivo

En este apartado explicaremos como acceder a los datos de un dispositivo, no trataremos las herramientas que permiten el clonado del dispositivo con iPhone DataProtection, pero si comentaremos las que se realizan mediante un backup de iTunes o simplemente accediendo desde un cliente SSH.

En el caso que el terminal objetivo tenga hecho el Jailbreak y tenga instalado OpenSSH, probaremos a conectarnos con cualquier cliente SSH con el usuario *root* y la contraseña *alpine*, que es la que viene por

defecto. Hay que tener en cuenta que el dispositivo tiene que estar conectado a la red Wifi para poder realizar la conexión. En caso que no podamos conectarnos porque el servidor SSH no esté abierto en el puerto por defecto, habrá que localizarlo utilizando alguna herramienta de escaneo de puertos como NMAP.

Una vez localizado el servidor SSH, como acabamos de comentar, probaremos con el usuario *root* y la contraseña *alpine* o con el usuario *mobile* y la contraseña *dottie*.

En el caso que el usuario haya cambiado las contraseñas y no lográramos acceder, se podría intentar un ataque de fuerza bruta o de diccionario contra este servicio.

También se podría dar el caso que el terminal no estuviera conectado a la red wifi. En este caso podríamos intentar realizar una conexión mediante cable y utilizando la técnica de USBMUX. Esta técnica utiliza el cable USB para realizar una conexión mediante SSH.

Por otra parte, si tenemos un dispositivo del que conocemos el passcode o este está desactivado, tendremos total acceso a este y podremos conectarlo y emparejarlo a un equipo para acceder a los datos, incluso sin la necesidad de realizar el proceso de Jailbreak. Pero en el caso que el passcode esté activado y lo desconozcamos, el dispositivo no tenga el Jailbreak y necesitemos acceder a los datos, tendremos que realizar el Jailbreak. Y en el caso que la versión de iOS del aparato no tenga la posibilidad de realizar un Jailbreak sin passcode, nos quedarán tres opciones. Averiguar el passcode o intentar saltárselo mediante algunas técnicas conocidas, obtener un equipo donde el terminal esté pareado o darnos por vencidos e intentar sacar el máximo de información del terminal mediante Siri y realizando análisis del tráfico de red en el caso que estuviera conectado a alguna y transmitiendo.

A continuación explicaremos algunas de las técnicas disponibles para desbloquear la pantalla de inicio sin conocer el passcode y otras para intentar obtenerlo.

Técnicas para obtener el passcode:

Smudge attack: Al utilizar el dedo para desbloquear un dispositivo por pantalla se produce un contacto en el cual se transmite la grasa corporal al dispositivo. Se puede analizar esta grasa corporal por medio una fotografía de calidad utilizando la técnica “Smudge Attack” y averiguar el código de desbloqueo mediante las marcas en la pantalla del dispositivo.

Exploit Limera1n: Si el terminal tiene un chip A4, es decir, si es un iPhone 4 o iPad 1 entonces se puede utilizar Gecko, una herramienta que permite, haciendo un tethering jailbreak temporal, lanzar un ataque de fuerza bruta en el mismo terminal y obtener el passcode. Se basa en iPhone-DataProtection y tiene el inconveniente de que puede ser un proceso lento.

Bug del teclado virtual en iPad utilizando Teensy 3.0: En iPad, al poner un teclado externo, cuando se fallan varias veces sólo se deshabilita el teclado virtual, pero no el físico. Con un Teensy 3.0 se puede hacer una emulación de un teclado y hacer un fuerza bruta para sacar el passcode. Hay que tener en cuenta que se podrían borrar los datos del terminal después de los 10 intentos. Afecta a todas las versiones de iOS hasta iOS 6.1.3 y todas las versiones de iPad.

Técnicas de evasión para saltarse el passcode:

Falsificación de huellas dactilares: El terminal iPhone 5S permite liberar el passcode y la contraseña de Apple ID haciendo uso de un sensor biométrico de huellas dactilares. Miembros del Chaos Computer Club han demostrado que se puede saltar con una huella dactilar falsa copiada de la víctima.

Bug en el Control Center de iOS 7.0: Se puede saltar el passcode accediendo al Control Center con el dispositivo bloqueado. Pulsando para apagar el terminal y después cancelando la operación al tiempo que se pulsa dos veces en el Home Button.

Bug de SmartCover en iPad en iOS 5.0: Un fallo grave que permite abrir la app que estuviera en primer plano. Se trata de un sencillo truco que consiste en dar a apagar el terminal iPad con el botón, pero antes de aceptar el apagado cerrar la SmartCover, volver a abrirla, y dar al botón de cancelar el apagado.

A modo esquemático y para simplificar el procedimiento hemos realizado un diagrama con los pasos a seguir. Seguidamente haremos una descripción concreta de cada fase del diagrama con sus posibles casos y excepciones que podemos encontrar.

En la siguiente tabla sintetizaremos los posibles casos que se podrán dar y el resultado, dependiendo de las condiciones de cada uno.

	Passcode	Jailbreak	Chip A4	Resultado
Caso 1	x	—	—	Acceso permitido
Caso 2	√	x	x	Acceso denegado
Caso 3	√	x	√	Acceso permitido(BootRom)
Caso 4	—	√	—	Acceso permitido(SSH o USB)

En esta fase comprobaremos si el dispositivo a analizar tiene activado el código de seguridad(passcode). En caso negativo ya tendríamos acceso a todas las funcionalidades del dispositivo, pero en caso afirmativo tendremos que resolver este problema. Para ello utilizaremos alguna de las técnicas que describimos en el apartado anterior, que hace referencia a la obtención y evasión del passcode. Dependiendo de la versión de iOS que tengamos, podremos utilizar de forma efectiva alguna de estas técnicas, y en caso contrario continuaremos con las comprobaciones en la siguiente etapa.

Si el dispositivo que tenemos tiene el passcode activado y lo desconocemos, tendremos que intentar averiguar si este tiene el Jailbreak realizado. Si desconocemos si el usuario del móvil tiene hecho el Jailbreak, tendremos que averiguar si este dispositivo tiene la app Cydia instalada y/o el servidor OpenSSH activo. Este último se suele instalar para realizar el Jailbreak y algunas herramientas para realizar este proceso lo dejan instalado, activado y con el password por defecto. Por tanto intentaremos acceder mediante SSH, tal y como hemos comentado anteriormente.

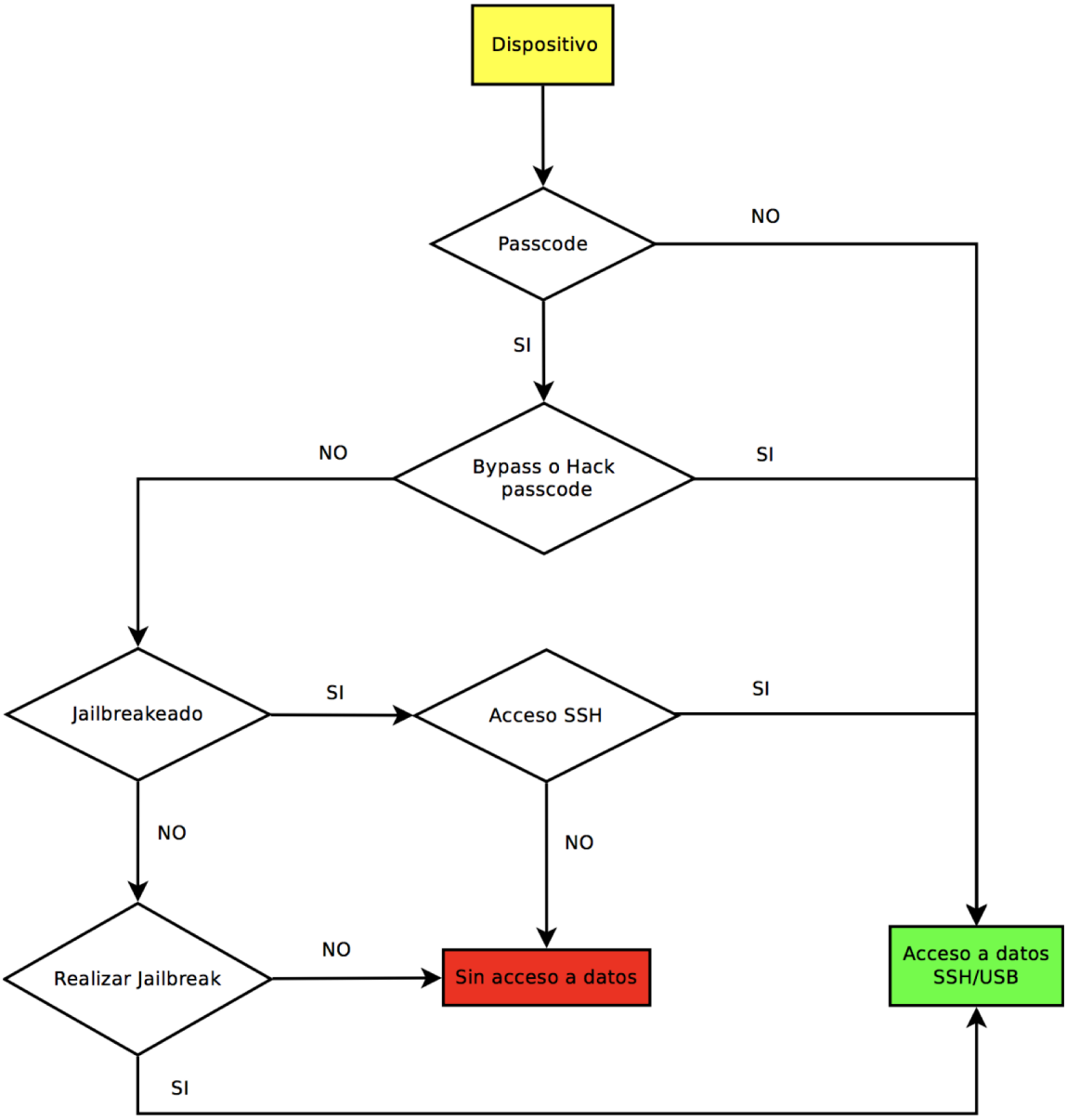


Diagrama con los pasos a seguir para obtener acceso a los datos.

En el caso que el resultado haya sido satisfactorio ya podremos acceder a los datos del dispositivo con cualquier cliente SSH y obtener los datos de este. En caso contrario tendremos que intentar realizar el Jailbreak sobre el dispositivo para poder acceder a los datos. Para saber si podemos realizar el Jailbreak sobre nuestro dispositivo, necesitaremos conocer la versión de iOS y el modelo en cuestión. Sino tendremos que ir probando las diferentes aplicaciones para realizar este proceso. Con estos datos consultaremos las tablas que hemos comentado anteriormente en la siguiente web, donde tendremos indicado la herramienta a utilizar para cada versión/modelo, en caso que sea posible:

https://www.reddit.com/r/jailbreak/wiki/escapeplan/guides/jailbreakcharts

Llegado a este punto podemos encontrarnos con diferentes casos:

1. Que para nuestro modelo y versión exista una herramienta de Jailbreak que no necesite el passcode.

2. Que nuestro dispositivo tenga el chipset A4 y por lo tanto es posible realizar el Jailbreak mediante el proceso BootRom.

3. Que no exista el Jailbreak para nuestro dispositivo.

En los 2 primeros casos, podremos conseguir hacer el Jailbreak y por lo tanto acceder a los datos, y en el caso 3 no podremos acceder. Solo podríamos intentar sacar el máximo de información del terminal mediante Siri, en el caso que esté activado, y realizando análisis del tráfico de red en el caso que estuviera conectado a alguna red Wifi y transmitiendo, como hemos comentado anteriormente.

Configuraciones del dispositivo y recomendaciones

En esta parte haremos un análisis de los controles de acceso y privacidad del sistema operativo iOS, así como de los servicios activos innecesarios y errores de configuración.

La seguridad de iOS respecto al acceso a los datos de las aplicaciones, se basa en un mecanismo muy básico de permisos que impone restricciones en las operaciones que un proceso puede llevar a cabo sobre ciertos elementos. Los elementos críticos de la plataforma que requieren de permisos para su acceso debido a que conllevan implicaciones desde el punto de vista de la privacidad del usuario, son definidos por Apple y varían en función de la versión de iOS empleada. Para ello, iOS solicita permiso al usuario a través de mensajes y ventanas gráficas de diálogo en el momento de hacer uso por primera vez de las funciones que requieren acceso a ciertos componentes, como por ejemplo la información de localización (GPS).

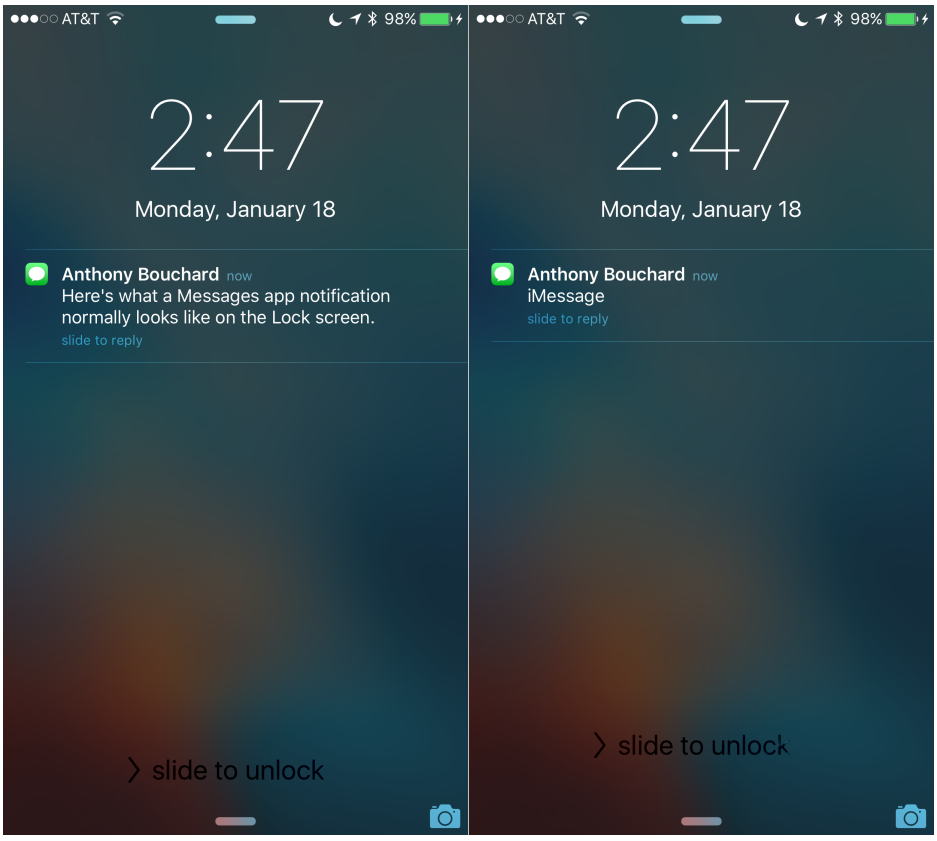
Aunque el número de permisos o controles de acceso se ha ido incrementando en las últimas versiones de iOS, solo se disponía de controles de acceso para aquellos permisos definidos en la plataforma, quedando otros permisos sin gestionar adecuadamente. A partir de iOS 7 extendió los controles de privacidad para gestionar el acceso al micrófono, el seguimiento a través de la publicidad o anuncios, la localización, etc.

Debemos prestar especial atención a las apps que disponen de acceso al micrófono, ya que podrán a través del mismo, escuchar todo el sonido y audio existente alrededor del dispositivo móvil, grabarlo y/o enviarlo a través de Internet.

Adicionalmente, iOS 7 permite desde la sección de publicidad restablecer el identificador de publicidad empleado para ofrecer anuncios personalizados según las preferencias o intereses del usuario, generando un nuevo valor aleatorio y temporal, no asociado al dispositivo móvil, al usuario, o al valor existente previamente. Desde el punto de vista de la privacidad, se recomienda restablecer periódicamente este identificador para que terceros no puedan realizar un seguimiento detallado del usuario a través de las redes de anuncios y publicidad. Complementariamente, se recomienda deshabilitar que se envíe información de la localización del usuario en las comunicaciones de anuncios de Apple Ads. Para ello se debe deshabilitar el acceso a la localización por parte de Apple Ads desde el menú "Ajustes → Privacidad → Localización → Servicios del sistema" y la opción "Apple ads según la ubicación".

La posibilidad de disponer de acceso físico al dispositivo móvil permitiría a un potencial atacante acceder a los contenidos del mismo o hacer uso de los servicios de telefonía móvil disponibles. Para evitar ambos tipos de acceso es posible fijar un PIN o código de acceso tanto en la tarjeta SIM como en el propio dispositivo móvil. El PIN de la tarjeta SIM bloquea el acceso no autorizado a los servicios y capacidades de telefonía móvil asociados a la propia tarjeta SIM. Si la tarjeta SIM está bloqueada, únicamente se permite la realización de llamadas de emergencia. Sin embargo, si sólo se establece el PIN en la tarjeta SIM y no en el dispositivo, aún es posible para un potencial atacante acceder al terminal directamente, incluyendo sus datos y aplicaciones, incluso sin extraer el SIM del dispositivo móvil.

iOS ha presentado a lo largo del tiempo y de sus múltiples versiones numerosas vulnerabilidades en la pantalla de desbloqueo, protegida por un código de acceso, tal y como mencionamos en apartados anteriores. Por lo tanto se recomienda tener el dispositivo y sus aplicaciones actualizados a la última versión. Además existen funcionalidades de Apple que podrían suponer una fuga de información, aún con el dispositivo protegido con código de acceso, como por ejemplo el sistema de reconocimiento de voz llamado Siri. Siri permite acceder a parte de la funcionalidad del teléfono mediante comandos de voz, y por ejemplo, enviar mensajes de texto o e-mails, realizar llamadas, y acceder al calendario o a los contactos. Se recomienda por tanto deshabilitar la



Captura de la pantalla de bloqueo de un dispositivo iPhone.

funcionalidad de Siri cuando el dispositivo móvil está bloqueado, pudiendo hacer uso de Siri tras desbloquear el terminal.

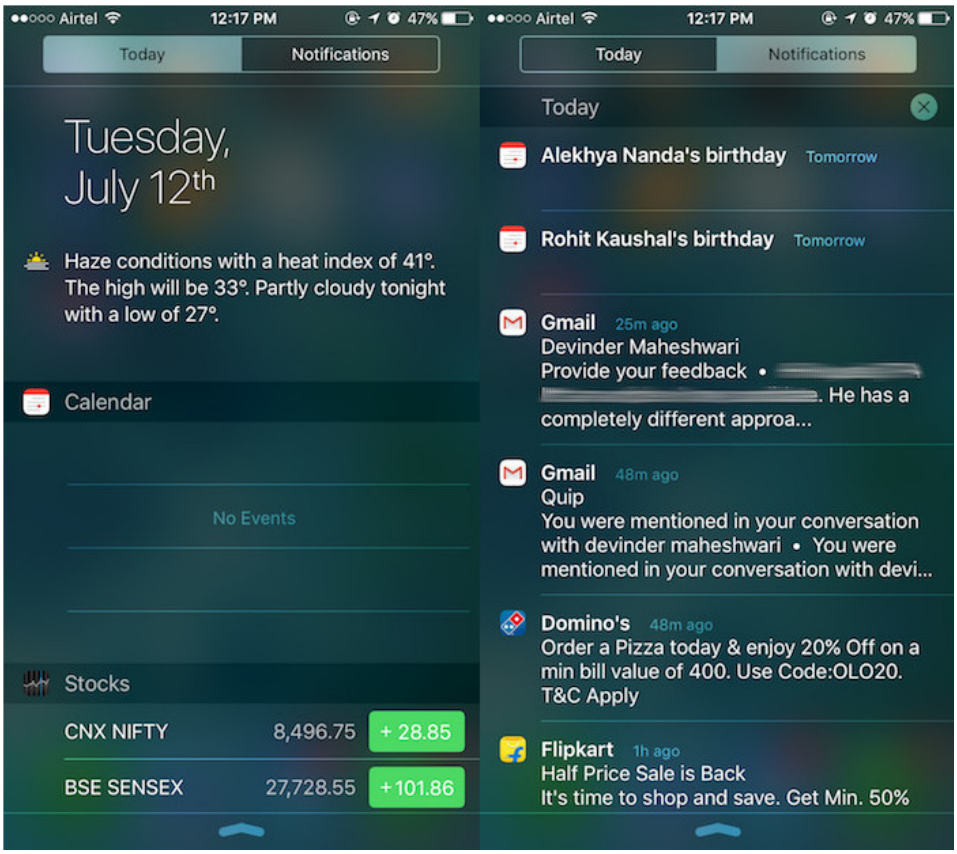
La pantalla de desbloqueo del terminal en iOS permite, sin necesidad de introducir el código de acceso y, por tanto, sin necesidad de desbloquear el teléfono, junto a la hora y la fecha, la previsualización del nombre del operador de telecomunicaciones y de la barra de estado, que incluye los iconos de conectividad y de notificaciones.

Adicionalmente, la pantalla muestra las llamadas de teléfono recibidas, siendo posible visualizar el remitente e incluso devolver la llamada, sin conocer el código de desbloqueo, deslizando el dedo sobre la notificación de llamada perdida, con las implicaciones de seguridad asociadas que tiene el poder realizar una llamada desde un dispositivo móvil sin autorización, suplantando a su usuario.

También la pantalla de previsualización permite visualizar los mensajes de texto y otras aplicaciones de iOS, como recordatorios, calendario, emails, etc.

La pantalla de desbloqueo puede desvelar información confidencial a un potencial atacante, ya que esta funcionalidad de previsualización podría permitir a un potencial atacante la lectura de información sensible simplemente con poder tener acceso visual a la pantalla del dispositivo.

Por tanto se recomienda desactivar las notificaciones con pantalla bloqueada de las aplicaciones, desde el menú “Ajustes → Notificaciones”.



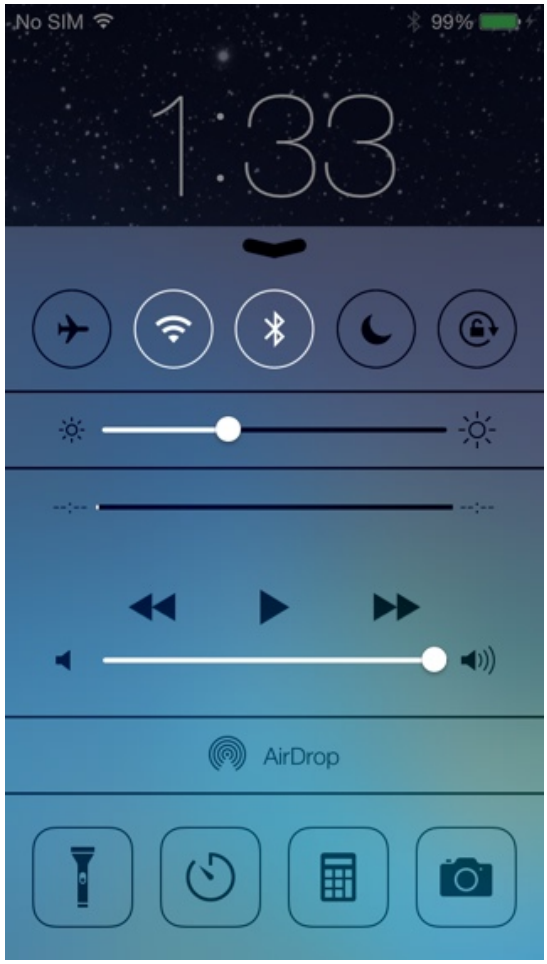
Capturas de pantalla de un terminal iPhone, con la pantalla bloqueada.

Por defecto en el caso del iPhone, iOS dispone de dos entradas correspondientes a la app Passbook, y a la posibilidad de responder la recepción de una llamada con un mensaje. Ambas opciones están activas por defecto. Desde el punto de vista de seguridad, si no se desea que un tercero pueda disponer de acceso a Passbook o responder con un mensaje, se recomienda deshabilitar ambas opciones.

El Centro de Notificaciones ya existía en iOS desde la versión 5 como un lugar centralizado para la obtención de mensajes y notificaciones de la plataforma y de las apps, y con capacidades para gestionar qué aplicaciones podían mostrar notificaciones y en qué formato. El nuevo Centro de Notificaciones introducido por Apple en iOS 7 permite disponer de más información, incluso desde la pantalla de desbloqueo, y configurar más aspectos asociados a su funcionamiento. El Centro de Notificaciones desde la versión 7 de iOS está accesible al desplazar hacia abajo la parte superior de la pantalla de iOS, por defecto tanto en la pantalla de bloqueo como una vez el dispositivo móvil ha sido desbloqueado.

Desde el punto de vista de seguridad, en el caso de gestionar información sensible y confidencial, se recomienda deshabilitar el centro de notificaciones mediante el menú "Ajustes → Centro de notificaciones".

Complementariamente al Centro de Notificaciones descrito previamente, desde iOS 7 se dispone también del Centro de Control, funcionalidad que permite disponer de accesos directos a los ajustes de configuración más frecuentemente utilizados por los usuarios, incluso desde la pantalla de desbloqueo. El Centro de Control está accesible al desplazar hacia arriba la parte inferior de la pantalla de iOS, por defecto tanto en la pantalla de bloqueo como una vez el dispositivo móvil ha sido desbloqueado. El Centro de Control ofrece controles para la reproducción de audio y vídeo, el volumen, acceso a AirDrop y a AirPlay, y el brillo de la pantalla. Adicionalmente, en su parte inferior cuenta con función de linterna, acceso al reloj, a la calculadora y a la cámara de fotos. En su parte superior proporciona controles para habilitar o deshabilitar el modo vuelo o modo avión, el interfaz Wi-Fi, el interfaz Bluetooth, el modo para no ser molestado y el bloqueo de la orientación de la pantalla.



Captura de pantalla del centro de control

Desde el punto de vista de seguridad es crítico habilitar el Centro de Control si la pantalla está bloqueada, ya que en caso contrario un potencial atacante con acceso físico al dispositivo móvil podría fácilmente, activar el modo avión o activar o desactivar cualquiera de los otros interfaces de comunicaciones Wi-Fi, Bluetooth, etc. Por lo tanto se recomienda deshabilitar la posibilidad de acceder al Centro de Control desde la pantalla de desbloqueo del dispositivo móvil mediante el menú "Ajustes → Centro de

control".

La principal recomendación de seguridad asociada a las comunicaciones Wi-Fi (802.11) en dispositivos móviles es no activar el interfaz inalámbrico Wi-Fi salvo en el caso en el que se esté haciendo uso del mismo, evitando así la posibilidad de ataques sobre el hardware del interfaz, el driver o la pila de comunicaciones Wi-Fi. iOS proporciona información en la barra superior de estado sobre si se está o no conectado a una red Wi-Fi actualmente. Hasta que el dispositivo móvil no es desbloqueado, únicamente genera tráfico de comprobación de las redes Wi-Fi visibles, y una vez es desbloqueado establece la conexión con la red Wi-Fi de preferencia.

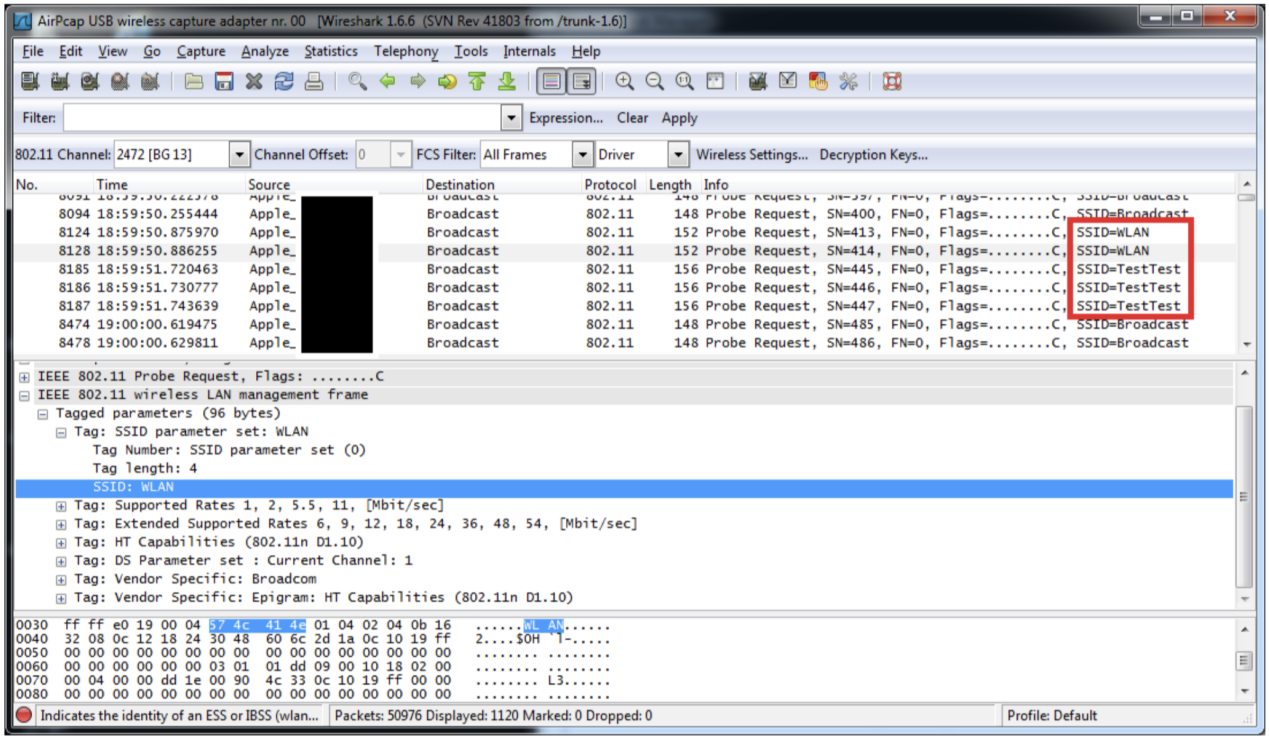
Por defecto, iOS se conecta de forma automática a las redes conocidas, no disponiéndose de ninguna opción para deshabilitar este comportamiento. La opción "Preguntar al conectar" (habilitada por defecto) establece si iOS preguntará al usuario en caso de necesitar disponer de una conexión de datos y no detectar la existencia de una actualmente, por ejemplo, porque una aplicación genere tráfico de datos. Al identificarse que es necesario disponer de una conexión y no existir ninguna activa, por ejemplo al no existir ninguna red Wi-Fi conocida en el alcance del dispositivo móvil, tras generar la notificación de datos móviles desactivados, iOS muestra al usuario una lista de redes Wi-Fi disponibles en la ubicación actual, y le pregunta si desea acceder a una red Wi-Fi abierta nueva dentro del alcance.

Una limitación de iOS relevante desde el punto de vista de seguridad es que sólo permite eliminar una red Wi-Fi si actualmente el dispositivo móvil se encuentra en su rango de alcance o cobertura, no siendo posible en otro caso. Como resultado, numerosas redes Wi-Fi permanecerán en la PNL y no podrán ser eliminadas por el usuario, ya que el dispositivo no volverá a estar bajo su rango de alcance, y el usuario desconocerá en muchos casos que las mismas residen en la configuración WiFi interna del dispositivo móvil.

En iOS disponemos de una opción para añadir manualmente la configuración de una nueva red Wi-fi a través del botón "Otra..." dentro de "Ajustes → WIFI". Pero no se recomienda añadir ninguna red Wi-Fi manualmente al dispositivo móvil a través del botón "Otra...", aunque es una práctica común desde el punto de vista de seguridad para disponer de mayor control sobre la configuración de la red Wi-Fi, ya que iOS generará tráfico que desvelará estas redes Wi-Fi disponibles en su lista de redes preferidas e intentará conectarse a ellas, escenario que puede ser empleado por un potencial atacante para disponer de conectividad con el dispositivo móvil y proceder a explotar vulnerabilidades en el mismo.

iOS dispone de capacidades nativas para actuar como punto de acceso Wi-Fi y router hacia Internet (función de Compartir Internet), proporcionando acceso a múltiples dispositivos u ordenadores vía Wi-Fi a su conexión de datos de telefonía móvil. A través del botón "Compartir Internet", y teniendo el interfaz Wi-Fi activo en iOS, es posible habilitar el servicio de punto de acceso Wi-Fi, de forma que otros ordenadores y dispositivos puedan utilizar el dispositivo móvil como punto de acceso Wi-Fi para conectarse a Internet.

En este aspecto se recomienda no habilitar las capacidades de punto de acceso Wi-Fi en el dispositivo salvo que se quiera hacer un uso limitado y controlado de esta funcionalidad. En caso de ser utilizada, la red Wi-Fi asociada debe ser protegida y configurada mediante una contraseña o clave precompartida suficientemente larga. En caso de activar el punto de acceso Wi-Fi, se recomienda también modificar el nombre de la red Wi-Fi (o SSID) anunciada por el dispositivo móvil por defecto (nombre del propio dispositivo móvil), con el objetivo



Captura del tráfico de red de un terminal iPhone, cuando se intenta conectar a una red Wifi.

de no desvelar que se trata de un punto de acceso basado en un dispositivo móvil iOS o datos de su propietario. Sin embargo, debe tenerse en cuenta que las tramas beacon del punto de acceso creado por iOS incluyen etiquetas propietarias de Apple que desvelan, al menos, el fabricante del dispositivo.

A partir de iOS 7 se introduce soporte para AirDrop, previamente sólo disponible en OS X 10.7.x (Lion) o superior, integrando en esta plataforma móvil las capacidades de transferencia de datos inalámbricas, incluyendo el intercambio de ficheros, fotos, vídeos, sitios web, localizaciones, etc. AirDrop emplea tanto tecnologías Wi-Fi como Bluetooth, debiendo ambos interfaces estar activos para poder hacer uso de las capacidades de AirDrop. Pese a hacer uso de TLS, es potencialmente vulnerable en ciertos escenarios a ataques de interceptación o Man-in-the-Middle (MitM), ya que puede emplear certificados anónimos disponibles en la librería CFNetwork de Apple. Por este motivo, se recomienda incrementar la concienciación del usuario respecto a esta tecnología, si se hará uso de ella, y en qué escenarios.

En cuanto a las comunicaciones bluetooth, la principal recomendación de seguridad en dispositivos móviles es no activar el interfaz inalámbrico Bluetooth salvo en el caso en el que se esté haciendo uso del mismo, evitando así la posibilidad de ataques sobre el hardware del interfaz, el driver o la pila de comunicaciones Bluetooth, incluyendo los perfiles Bluetooth disponibles.

Hay detalles que no se tienen en cuenta a la hora de configurar un dispositivo nuevo. El dispositivo móvil está configurado por defecto con valores fijados por el fabricante del sistema operativo y del hardware, y ligeramente por el operador de telefonía móvil en función del perfil de configuración de iOS asociado a la tarjeta SIM. Estos valores pueden desvelar detalles del terminal o de su propietario. Para evitar la revelación de

información sensible, se recomienda modificar esos valores existentes por defecto por valores elegidos por su propietario. Adicionalmente, durante el proceso de configuración inicial iOS solicita al usuario la posibilidad de remitir a Apple información de diagnóstico y uso del dispositivo móvil, y se recomienda no permitir el envío de dicha información a Apple para evitar revelar detalles privados del uso y la ubicación del dispositivo móvil.

Si vamos a la configuración de Safari, el navegador web por defecto de Apple, a través del menú “Ajustes → Safari”, es posible acceder a los ajustes de configuración del navegador web. En este apartado se recomienda no habilitar las opciones de “Contraseñas y autorrelleno” y “Tarjetas de crédito”, dejar activada la opción “Bloquear cookies”, “Aviso de sitio web fraudulento” y”No rastrear”. Para una navegación web más segura, pero muy restringida desde el punto de vista de su funcionalidad, se recomienda deshabilitar la ejecución de scripts mediante la opción "JavaScript", y el uso de cookies en el navegador web (mediante la opción "Bloquear cookies: Siempre"), con el objetivo de mitigar los ataques asociados a la ejecución de scripts maliciosos en el navegador web y a la privacidad del usuario a través de las cookies. Adicionalmente, el bloqueo de ventanas o pop-ups es una opción que debe permanecer habilitada.

La aplicación cliente disponible por defecto, "Mail", para la gestión de correos electrónicos en iOS presenta diferentes debilidades de seguridad que deben ser tenidas en cuenta por los usuarios y organizaciones. Por defecto, la aplicación tiene habilitada la carga de imágenes contenidas en correos electrónicos en formato HTML. Esta funcionalidad permitiría a un potencial atacante enviar a un usuario víctima un correo electrónico con una imagen incluida y automáticamente el dispositivo móvil iOS al proceder a la visualización del e-mail, solicitaría la imagen. También es recomendable modificar la firma por defecto para no revelar información del fabricante y del dispositivo móvil, ya que aparece la firma "Enviado desde mi iPhone".

Por otro lado, también debemos tener en cuenta qué aplicaciones y servicios tenemos activados y no son necesarios. En el caso de las aplicaciones, sobretodo si tenemos el dispositivo con el Jailbreak, debemos evitar instalar apps que no tengamos contrastadas y que nos puedan hacer sospechar sobre su verdadero funcionamiento.

Malware en iOS

A pesar de que un terminal con iOS cuenta con muchas protecciones contra apps no deseadas en la AppStore, debido a las soluciones implementadas por Apple Inc., y de que el sistema está limitado a usuarios no privilegiados por defecto, siguen existiendo posibilidades y formas de ser infectado por un malware.

Se pueden utilizar diferentes métodos para realizar esta infección, a continuación explicaremos algunos. Para algunos de ellos las técnicas de ingeniería social a utilizar serán muy importantes para que la infección tenga éxito.

Si el terminal objetivo tiene hecho el Jailbreak y utiliza Cydia, existen varias formas de instalarle el malware al terminal. La primera de ella sería posible instalando el malware desde algún repositorio usando Cydia. Para ello habría que subir la app a Cydia u otro repositorio y convencer al usuario que instale esa aplicación con trucos de ingeniería social. Se podría introducir el código malicioso, por ejemplo, dentro de un juego.

Archivos	Localización
Historial de Llamadas	\private\var\mobile\Library\CallHistory\
Contactos	\private\var\mobile\Library\AddressBook\
SMS	\private\var\mobile\Library\SMS\
Calendario	\private\var\mobile\Library\Calendar\
Notas	\private\var\mobile\Library\Notes\
Email	\private\var\mobile\Library\Mail\
Grabaciones de voz	\private\var\mobile\Media\Recordings
Apps instaladas	\private\var\mobile\Applications
Fuentes de Cydia	\var\lib\apt\list
Fotos	\private\var\mobile\Media\DCIM\100APPLE

Muchos repositorios de apps con Jailbreak no son tan escrupulosos en los controles y es posible subir una app con un troyano en ella. En los repositorios oficiales de Apple no es tan sencillo, pero se han dado varios casos que comentaremos más adelante.

A pesar de que el dispositivo móvil con iOS no tenga hecho el Jailbreak, si es posible convencer al usuario de que acepte un *provisioning profile**, será posible instalar un malware en el equipo si este va firmado digitalmente. Para poder realizar esta técnica, si se cuenta con un Apple Developer ID se puede firmar el código e instalarlo en el equipo aun no habiéndose hecho el Jailbreak.

* *Provisioning Profile:* es un fichero que asocia tu certificado de iOS con el App ID de tu app.

Utilizando las técnicas mencionadas anteriormente en el apartado "acceso a datos", con las que obtenemos acceso al dispositivo, podemos instalar el malware usando una conexión ssh, o en todo caso si se tiene acceso al terminal mediante cable, utilizando el puerto USB.

Se han dado algunos casos de apps maliciosas que se habían saltado todos los controles de Apple y se podían descargar del repositorio oficial de la AppStore. Un caso ampliamente conocido es el de la app Find and Call. Esta aplicación robaba agendas de contactos sin conocimiento ni consentimiento por parte del usuario. O el caso de uno de los malwares más conocidos y que tuvo mucha repercusión, sobretodo en China, fue XcodeGhost.

El denominado XcodeGhost es un troyano, un código malicioso que modifica el entorno de desarrollo integrado Xcode de Apple para infectar las aplicaciones que están siendo desarrolladas con él para iOS. La principal función de este malware es la de obtener información sobre los dispositivos en los que se instalan.

Al estar presente en el propio entorno de desarrollo de Apple, este troyano se ha camuflado en aplicaciones oficiales alojadas sobre todo en la versión china de la AppStore. Siendo capaz de hacer que estas desvíen a servidores de terceros diferentes tipos de información.

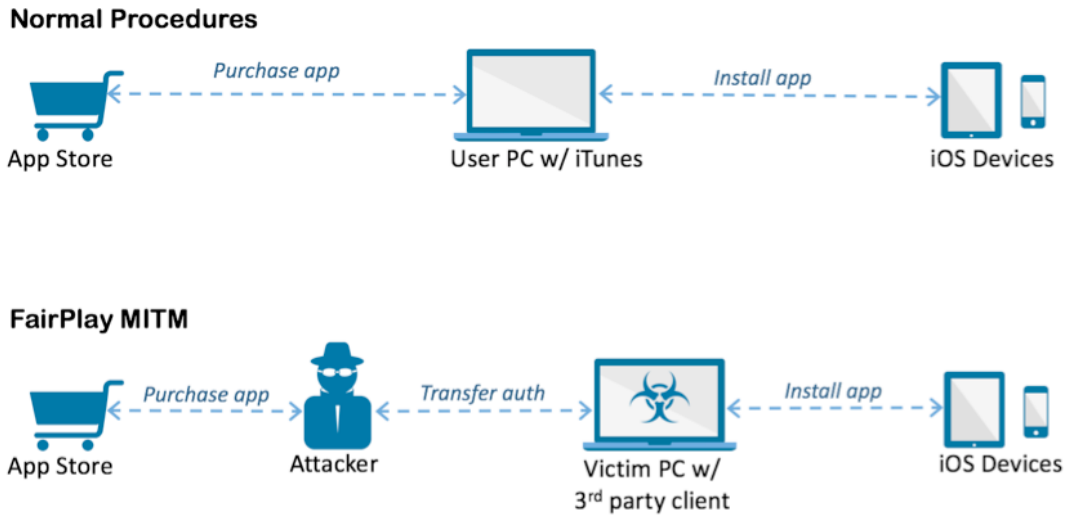


Diagrama del ataque de red FairPlay Man in the Middle que realiza AceDeceiver.

Un nuevo tipo de malware de iOS que explota defectos en el diseño de la gestión de derechos digitales por parte de Apple, se instala sin necesidad de tener el sistema con Jailbreak. A este malware se le denomina AceDeceiver y la técnica que utilizada es conocida como FairPlay Man in the Middle. Los usuarios que querían software pirata tienen un código de autorización para una aplicación legítima, como exige el protocolo FairPlay. Estos códigos son solicitados por los dispositivos iPhone desde iTunes para probar que la aplicación fue comprada. Los piratas utilizaron software para PC, el cual se hizo pasar por un cliente de iTunes para que se pudiera almacenar y enviar códigos de autorización para aplicaciones. Estos códigos podrían ser utilizados para engañar con eficacia a un dispositivo iOS.

Para el cliente fake de iTunes utilizaron una herramienta popular llama Aisi Helper Windows, que se ejecuta en el sistema operativo de Microsoft, pero no había sido utilizada previamente para este tipo de propósitos. Cuando los usuarios descargan Aisi, el sistema intentará automáticamente descargar el malware de la AppStore y la instalará automáticamente en los dispositivos iOS adjuntos. No se requiere confirmación por parte del usuario. Una vez que se conseguía instalar apps los atacantes comenzaron con el desvío de información de los usuarios infectados.

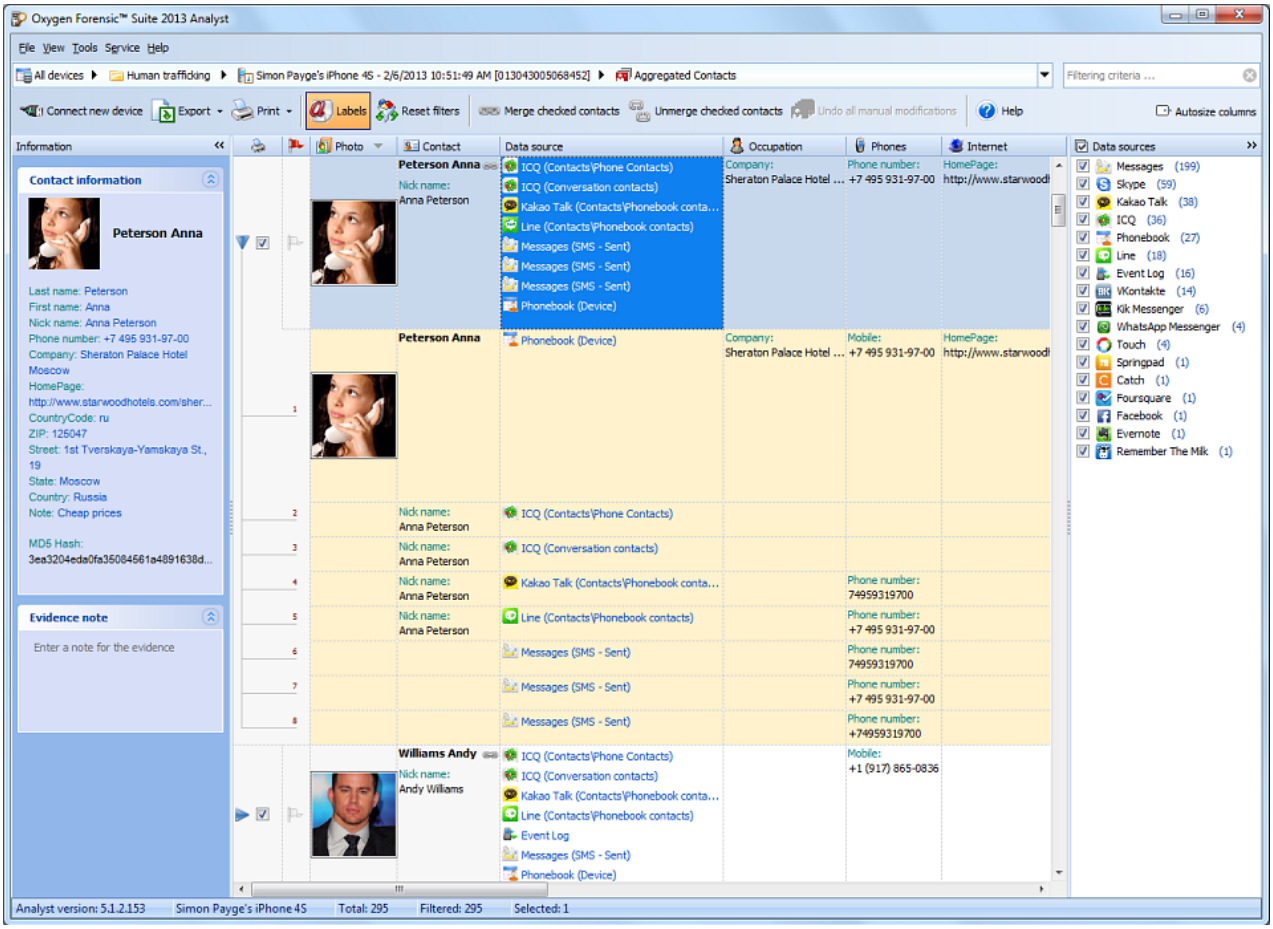
Por último queremos comentar la técnica *address bar spoofing* que se basa en la ingeniería social, engañado al usuario a través del navegador. Utiliza una vulnerabilidad de la barra de direcciones del navegador, que afecta a algunas versiones de iOS. Se trata de engañar al usuario haciéndole creer que se encuentra en un sitio web, cuando en realidad de encuentra en otro. Es una técnica utilizada en ciertos ataques de phishing para robar credenciales u otro tipo de información del usuario.

Análisis de datos

Estructura del sistema de archivos

Para acceder a los datos del dispositivo, ya sea por ssh o por usb, podemos utilizar alguna de las herramientas tales como iFunbox, DiskAid o iExplorer, que nos permitirán inspeccionar la estructura del terminal de manera cómoda y sencilla.

Dependiendo si el terminal tiene hecho el Jailbreak o no, podremos acceder al sistema de ficheros completo o solo a una gran parte de él. Cada directorio tiene sus permisos y los archivos que contiene poseen sus permisos propios, que pueden ser diferentes. Los principales archivos del núcleo del sistema operativo se



Listado de los contactos obtenidos con Oxygen Forensic Suite.

encuentran en el directorio: \System\Library. Los datos se almacenan en el directorio: \private\var, es como una especie de partición diferente. Los datos del usuario, están en el directorio: \private\var\mobile.

En la tabla anterior se presentan algunas de las localizaciones de archivos en el sistema operativo iOS, con Jailbreak.

Evidentemente iFunBox o iExplorer no son como las herramientas de análisis forense profesionales, con las comodidades de pulsar un botón y listo, y tampoco pretende suplantar a una conexión terminal al dispositivo

para acceder a los ficheros. Pero puede servir para obtener información del dispositivo, de los datos y de las aplicaciones instaladas.

Análisis con Oxygen Forensic

En este último apartado vamos a explicar el procedimiento a realizar para la extracción análisis de los datos, mediante la herramienta de análisis forense Oxygen Forensic Suite.

Este procedimiento de análisis se podría realizar de forma manual, sin embargo el volumen de datos es tan grande que deberíamos pasar varias semanas con este proceso. Es por ello que para facilitar el trabajo haremos uso de Oxygen Forensic Suite, que nos permitirá analizar los datos de forma rápida y eficiente.

Oxygen Forensic Suite es una herramienta comercial para el análisis forense en telefonía móvil. Es compatible con la mayoría de los teléfonos del mercado, aunque esta solución se especializa en la extracción y análisis de datos en smartphones, apoyándose en una interfaz cómoda e intuitiva. Actualmente en su página web ya no se puede obtener una demo del producto, pero todavía existen páginas que ofrecen las últimas versiones que fueron gratuitas durante 30 días, con algunas limitaciones como el número de ejecuciones y la prohibición de utilizar la herramienta para fines comerciales.

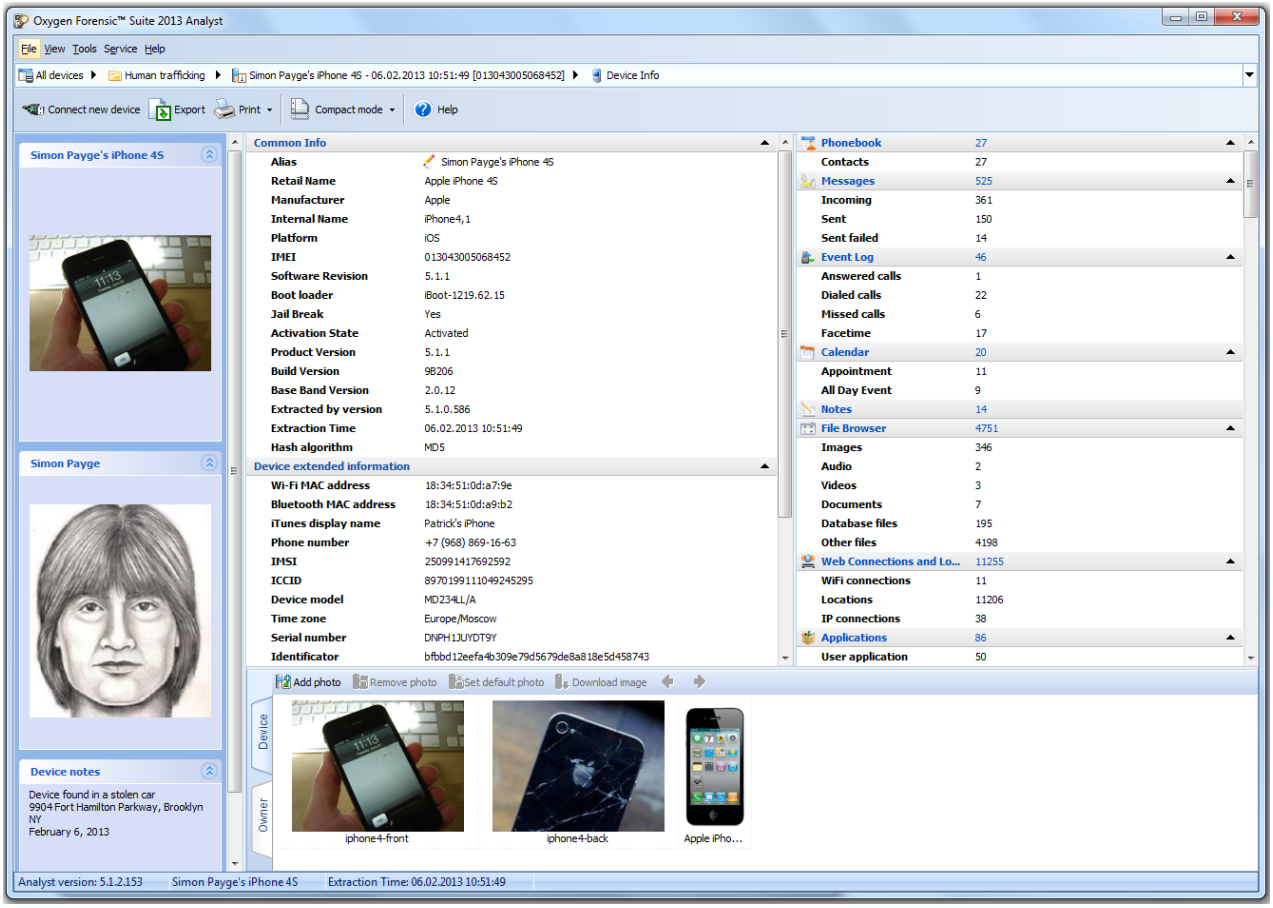
Una de las características importantes de esta aplicación, aunque no lo vayamos a tratar en este trabajo, es que permite extraer y analizar datos almacenados en los archivos de copia de seguridad hechas por iTunes o iCloud.

Para el proceso de captura de datos se recomienda utilizar el cable USB y drivers originales, proporcionados por el fabricante. Para realizar el proceso de extracción y análisis con Oxygen Forensic es necesario tener el equipo pareado al terminal previamente. Es decir, que necesitaremos que el terminal no tenga passcode o conocer el passcode para poder hacerlo.

El proceso de extracción de los datos puede ser largo, dependiendo del tamaño de la memoria del dispositivo a analizar. Una vez terminado el proceso, podemos desconectar el terminal y empezar a utilizar la herramienta para consultar los resultados obtenidos.

Lo primero que obtenemos es información técnica completa del dispositivo. Esto incluye el fabricante, el modelo, la plataforma y su versión, el número IMEI, la dirección MAC, el número de serie, el número de teléfono y cualquier otro dato específico del modelo. Esta sección también ofrece un resumen donde muestra las cuentas del usuario recopiladas a partir de todos los datos extraídos del dispositivo, así como el número de contactos, mensajes, calendario, notas y el número de archivos encontrados.

Esta herramienta también posee una opción de Time-Line, en la que saca todos los eventos temporales de todas las aplicaciones, desde que se enciende el móvil. Muestra los sucesos como recibir una llamada, enviar un email, publicar un Twit, hacer una fotografía, conectarse a una red Wifi, etc. Es decir, todo lo que hace el



Resumen de los datos obtenidos con Oxygen Forensic Suite.

terminal queda registrado en las aplicaciones del sistema y por tanto en la base de datos, ficheros de configuración, etc y Oxygen Forensic los analiza todos y los muestra en una línea temporal.

Lo siguiente que vamos a ver es el análisis de contactos. Los contactos se almacenan en una base de datos SQLite, esta contiene todos los datos relacionados con estos, como son el nombre, ocupación, número de teléfono, dirección, correo electrónico, notas, etc. Es interesante saber que Oxygen Forensic es capaz de reconocer las agendas de las demás aplicaciones e indicar las interacciones que han habido con sus contactos, y mostrar toda esta información de forma conjunta, pudiendo filtrar por tipo de comunicación, contacto, teléfono, etc.

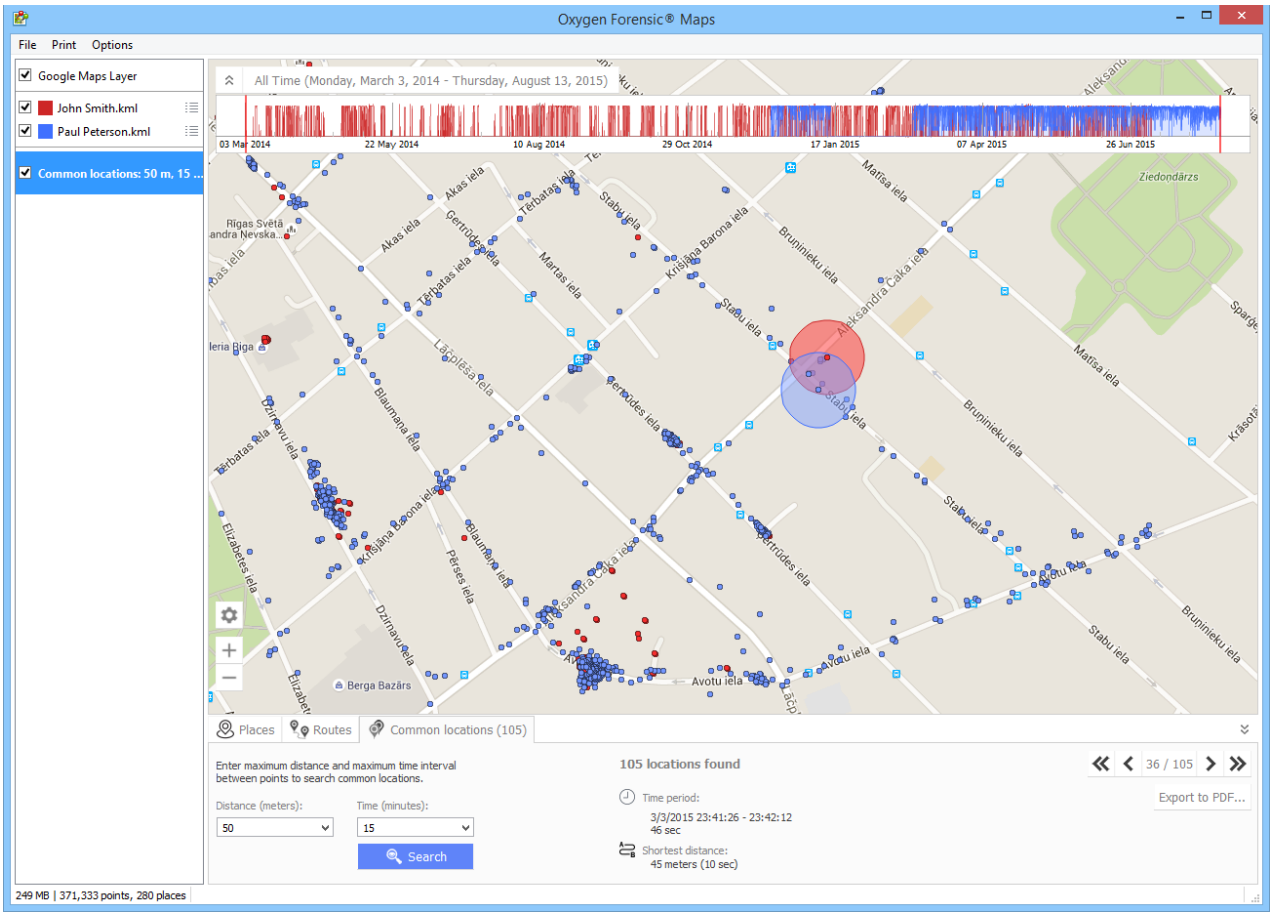
Además, el análisis de datos de estos contactos permite saber el grado de cercanía que un determinado usuario mantiene con el propietario. Esto se muestra a través de un gráfico circular de relaciones(“estadísticas de comunicación”), donde los más cercanos al núcleo son los que más relación tienen con él.

En la sección que contiene el registro de eventos podemos observar las comunicaciones de voz del usuario, donde además nos indica su tiempo, duración y el contacto con quien se realizó la comunicación. También es posible recuperar llamadas eliminadas.

Respecto a los mensajes de comunicación, en un smartphone podemos encontrar gran variedad de mensajería instantánea como Skype, Whatsapp, Messenger, Google Talk, etc, además de los canales de comunicación comunes como SMS/MMS, imessage y emails. Por tanto realizar un análisis de todas las aplicaciones con estas características implicaría analizar todas las bases de datos y ficheros de comunicación. Desde esta herramienta tenemos un panel especial donde aparecen todas las aplicaciones soportadas y desde donde podremos acceder a los registros analizados detalladamente de forma sencilla.

La sección del navegador de archivos es una herramienta poderosa que permite acceder a las fotografías, los videos, los documentos y las bases de datos del dispositivo y analizarlos. Los extractores de texto incorporado, valores hexadecimales, multimedia, SQLite, Plist Viewer, geolocalización y EXIF nos ayudan a visualizar los archivos y sus propiedades. Oxygen Forensic extrae automáticamente los datos de ubicación geográfica de archivos multimedia y ubica el lugar en el mapa. Además, analiza los encabezados EXIF para encontrar etiquetas específicas, como marca, modelo y marca de hora que ayudan a los expertos a determinar el origen del archivo.

Una característica destacada de esta herramienta es la sección de mapas. En esta funcionalidad, la aplicación adquiere las coordenadas geográficas de todos los medios posibles, y lo muestra en un mapa de forma detallada. De tal manera que podemos visualizar los lugares más visitados, los recorridos realizados y las fotos tomadas en cada lugar.



Mapas con las ubicaciones extraídas de los diferentes medios mediante Oxygen Forensic Suite.

En el caso del navegador web, podemos acceder la historial del navegador y sus cookies. Podremos extraer información como contraseñas guardadas en los diferentes sitios web, las páginas favoritas, sesiones activas e incluso datos de formularios que hayan sido cacheados por el servidor.

Para las aplicaciones de redes sociales más importantes como FaceBook, Twitter o FourSquare, también tendremos acceso a un análisis detallado donde para cada una de estas aplicaciones podremos ver las publicaciones, los amigos, los mensajes privados, e incluso una lista de los usuarios y contraseñas utilizados en cada una de estas aplicaciones.

En el caso que queramos obtener información sobre alguna de las aplicaciones instaladas, y la herramienta Oxygen Forensic no la tenga configurada para su análisis, podremos acceder a todos los ficheros de esta de forma manual y analizarlos por nuestra cuenta. Para revisar estos ficheros se acompañan visores para SQLite, ficheros Plist y Bplist, editores de texto y editores hexadecimales.

Y finalmente, otra de las características que pueden ser muy interesantes, es la posibilidad que nos ofrece esta herramienta de análisis forense para detectar aplicaciones spyware instaladas en el dispositivo, procesando sus registros y archivos de configuración. La presencia de spyware en el teléfono puede significar que las actividades del usuario del teléfono fueron monitorizadas o controladas.

Por supuesto, podremos generar diferentes informes de las evidencias obtenidas para su posterior exportación.

ESCANEO DE DISPOSITIVOS MÓVILES ANDROID

Estudio de las herramientas a utilizar

A continuación nombraremos el equipamiento necesario y algunas de las herramientas que utilizaremos en esta primera fase de escaneo y una breve descripción sobre ellas:

Equipo técnico:

- Ordenador(PC, Mac o Linux).
- Conexión a Internet.
- Cables de datos de enlace para dispositivos Android.

Herramientas necesarias:

- Nmap: Es una herramienta gratuita de escaneo de redes que permite identificar qué servicios se están ejecutando en un dispositivo remoto, así como la identificación de estos, y existencia de filtros o firewalls, entre otros.
- Wireshark: Wireshark es una herramienta multiplataforma open-source de análisis de red, producto de la evolución de Ethereal. Se utiliza para realizar análisis y solucionar problemas en redes de comunicaciones, efectuar auditorías de seguridad, para desarrollo de software y protocolos, y como una herramienta didáctica. Cuenta con todas las características estándar de un analizador de protocolos.
- Santoku Linux: Es una distribución de Linux basada en Lubuntu 14.04 especialmente diseñada y complementada con una serie de herramientas para llevar cabo tareas de Análisis de Malware, análisis forense de smartphones, y pruebas de seguridad de Aplicaciones. Posee una gran cantidad de herramientas ya preinstaladas y preconfiguradas para ayudarnos en futuras verificaciones y pruebas móviles. Entre ellas podemos destacar ExifTool, Iphone Backup Analyzer y SSL strip.
- Kali Linux: Es una distribución basada en Debian GNU/Linux diseñada principalmente para la auditoría y seguridad informática en general. Una de las principales virtudes de Kali Linux son las más de 300 herramientas y aplicaciones relacionadas con la seguridad informática que incluye esta distribución. Algunas aplicaciones mencionadas anteriormente las podemos encontrar en esta distro.

Exploración del dispositivo

Debido a la gran cantidad de fabricantes que disponen de modelos de smartphones que utilizan Android como sistema operativo móvil, no es una tarea sencilla identificar de forma visual el modelo exacto de un terminal Android.

Podemos averiguar el modelo exacto de terminal a través del código IMEI. Este código lo encontraremos en la mayoría de los smartphones Android en la parte trasera del dispositivo, detrás de la tapa y debajo de la

batería. La ubicación del número varía dependiendo del modelo de teléfono y del fabricante, pero usualmente está impreso en una pegatina que se encuentra debajo de la batería. Con este número de 15 dígitos podremos obtener la información técnica del dispositivo, haciendo la comprobación en http://www.imei.info.

Captura de la web donde identificar el modelo de smartphone. Fuente: www.imei.info

Es importante conocer el modelo de dispositivo porque esta es una información muy útil a la hora de rootear el terminal o firmware de forma manual. Ya que los archivos varían según el modelo del teléfono y debemos asegurarnos que la técnica a utilizar es la adecuada. Para evitar este tipo de inconveniente podremos utilizar herramientas de rooteo con compatibilidad global.

Indistintamente del modelo de dispositivo que dispongamos pasaremos a la fase de identificación del sistema de seguridad de acceso. Los dispositivos investigados en algunas circunstancias no poseen ningún sistema de seguridad para acceso a los datos, principalmente por descuido o desconocimiento de los usuarios. No obstante, existen dispositivos que protegen los datos generalmente con el cifrado o necesitan de métodos de autenticación usando una contraseña, u otro medio de acceso. Android implementa diferentes métodos de bloqueo de seguridad, algunos mediante funciones genéricas para todos los dispositivos y otras solo disponibles en dispositivos con un hardware concreto. A continuación describiremos algunas de ellas:

- Patrón de desbloqueo: Un patrón de desbloqueo es un código que consiste en unir una serie de puntos en una línea sin levantar el dedo de la pantalla, generando así un recorrido, que servirá como patrón.
- Código PIN: Es el método más conocido y el que utiliza la SIM para ser desbloqueada al activar el terminal. consiste en un código numérico de 4 a 17 dígitos que al ser introducido nos permitirá desbloquear el terminal.
- Contraseña: Posiblemente, este sea el método más seguro de los que vienen de serie en nuestros smartphones. Consiste en configurar una contraseña alfanumérica para poder desbloquear el dispositivo.
- Huella dactilar: Este sistema es el equivalente al Touch ID de iOS que implementa en los nuevos dispositivos Apple Inc. Es una característica exclusiva de fabricantes de terminales que poseen este tipo de sensor.

Dependiendo de la capa de personalización del fabricante y de la versión de Android un terminal puede tener configurado otro sistema de seguridad más complejo, a continuación describimos algunos:

- Dispositivos de confianza: Esta opción permite configurar diferentes aparatos Bluetooth como dispositivos de confianza. Esto significa que cuando este dispositivo esté vinculado al terminal, no requerirá de ningún tipo de contraseña o código para ser desbloqueado. Es muy útil cuando llevamos un wearable encima.
- Sitios de confianza: Este modo de configuración permite tener desbloqueado el dispositivo si estamos en un lugar previamente establecido, por ejemplo nuestra casa u oficina, y se apoya en la señal GPS para su funcionamiento.
- Cara de confianza: Con este método, el teléfono escaneará nuestro rostro gracias a la cámara frontal y detectará nuestros rasgos.
- Desbloqueo por voz: Esta opción nos permite desbloquear el teléfono con la voz,

En la actualidad, existen diferentes herramientas que proporcionan funciones automatizadas que permiten omitir los mecanismos de seguridad, así como recuperar contraseñas del dispositivo bloqueado. Se pueden citar herramientas de uso comercial como Oxigen Forensic Suite.

A la hora de identificar el dispositivo para poder detectar servicios o puertos abiertos, necesitaremos que el terminal esté conectado a una red Wi-fi conocida y que tengamos acceso. Para este tipo de escaneo necesitaremos hacer uso de la herramienta Nmap.

Uno de los primeros pasos a la hora de analizar la seguridad es comprobar los servicios visibles externamente, puesto que son la primera puerta ante cualquier ataque externo. La mejor forma para ello es

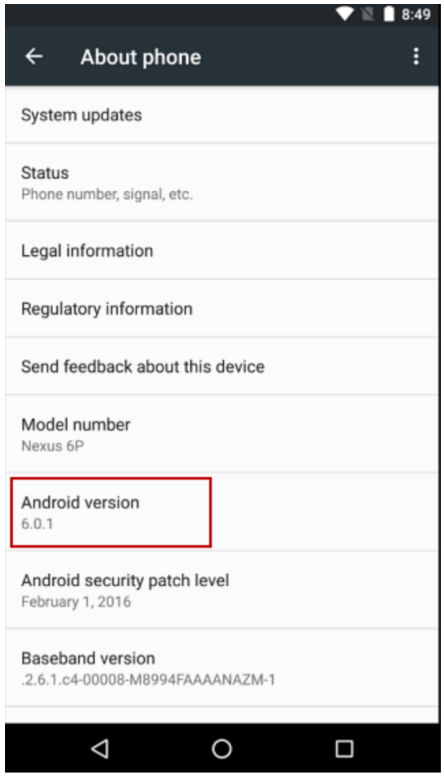
escanear todos los puertos del sistema, tanto puertos TCP como UDP. La herramienta Nmap es capaz de identificar si un dispositivo Android si tiene algún servicio en escucha.

Por defecto Android no tiene ningún puerto abierto. Y al poseer todos los puertos cerrados, cualquier escáner tiene mucho más complicado el poder reconocer el OS objetivo, en este caso Nmap no es capaz de especificarlo, todo lo contrario de lo que sucedería si escaneáramos un dispositivo iPhone, con su correspondiente sistema operativo iOS.

Análisis de la información obtenida

Llegado a este punto, si hemos podido distinguir el modelo de dispositivo y el fabricante a través de la pegatina que existe detrás de la batería, en algunos dispositivos, podremos llegar a saber qué herramientas podemos utilizar para realizar root en el dispositivo o utilizar alguna de las estándar.

En el caso que no hayamos podido averiguarlo y sí tengamos acceso al dispositivo porque conozcamos el código/patrón del sistema de desbloqueo, desde el menú “Ajustes -> Acerca del dispositivo” podremos ver la versión de Android que tiene instalada. Este menú puede encontrarse en un lugar distinto en el caso de tratarse de una versión de Android diferente.



Información del dispositivo en “Ajustes”.

Informe de resultados

En esta fase del estudio, después de haber explorado el dispositivo y analizado la información obtenida, debemos tener claro qué modelo de smartphone estamos estudiando, su versión de sistema operativo Android y en el mejor de los casos, si el dispositivo tiene algún servicio en escucha que posteriormente nos sirva para explotar alguna vulnerabilidad. Aunque esto último lo trataremos en la siguiente fase del estudio.

EVALUACIÓN Y ACCESO AL SISTEMA EN ANDROID OS

Estudio de las herramientas a utilizar

A continuación nombraremos el equipamiento necesario y algunas de las herramientas que utilizaremos en esta primera fase de escaneo y una breve descripción sobre ellas:

Equipo técnico:

- Ordenador(PC, Mac o Linux).
- Conexión a Internet.
- Cables de datos de enlace para dispositivos Android.

Herramientas necesarias:

- Tools para rootear dispositivos Android: Root Tools y Kingo Root son herramientas que permiten rootear un dispositivo Android, es decir obtener permisos de root para poder acceder a todo el contenido del dispositivo.
- Wireshark: Wireshark es una herramienta multiplataforma open-source de análisis de red, producto de la evolución de Ethereal. Se utiliza para realizar análisis y solucionar problemas en redes de comunicaciones, efectuar auditorías de seguridad, para desarrollo de software y protocolos, y como una herramienta didáctica. Cuenta con todas las características estándar de un analizador de protocolos.
- Santoku Linux: Es una distribución de Linux basada en Lubuntu 14.04 especialmente diseñada y complementada con una serie de herramientas para llevar cabo tareas de Análisis de Malware, análisis forense de smartphones, y pruebas de seguridad de Aplicaciones. Posee una gran cantidad de herramientas ya preinstaladas y preconfiguradas para ayudarnos en futuras verificaciones y pruebas móviles.
- Kali Linux: Es una distribución basada en Debian GNU/Linux diseñada principalmente para la auditoría y seguridad informática en general. Una de las principales virtudes de Kali Linux son las más de 300 herramientas y aplicaciones relacionadas con la seguridad informática que incluye esta distribución. Algunas aplicaciones mencionadas al inicio las podemos encontrar en esta distro.
- Oxygen Forensic Suite: Oxygen Forensic Suite es una herramienta comercial para el análisis forense en telefonía móvil. Es compatible con la mayoría de los teléfonos del mercado, aunque esta solución se especializa en la extracción y análisis de datos en smartphones, apoyándose en una interfaz cómoda e intuitiva.

Identificación de vulnerabilidades y Android rooting

Proceso de Rooteo

Antes de continuar con el siguiente tema debemos familiarizarnos con los términos y técnicas que vamos a tratar. Para ello debemos conocer el significado de rooting o rooteo, que nos ayudará a acceder a la totalidad de los datos de un smartphone con el sistema operativo Android.

"El rooting o rooteo de dispositivos Android es el proceso que permite a los usuarios de teléfonos inteligentes, tablets y otros aparatos con el sistema operativo móvil Android obtener control privilegiado (conocido como acceso root o permisos de superusuario). El rooting se lleva a cabo generalmente con el objetivo de superar las limitaciones que los operadores de telefonía móvil y los fabricantes de hardware colocan en algunos dispositivos, teniendo como resultado la capacidad de hacer cosas que un usuario sin root no puede hacer."
Fuente: es.wikipedia.org

La palabra root proviene del inglés y su traducción al castellano es "raíz". Ser usuario root en un sistema Unix como Android significa ser un Superusuario y tener, por lo tanto, acceso total al sistema.

Ser usuario root implica tener más permisos que el resto de usuarios. Por lo general, para tener estos permisos avanzados vamos a tener que modificar nuestro sistema. Todos los sistemas Android que los fabricantes incluyen en nuestros terminales no tienen acceso root por defecto para garantizar la seguridad del usuario.

Para rootear o hacer root a un terminal existen diferentes métodos. El proceso para ser root depende del fabricante, del modelo de nuestro dispositivo y de la versión Android que está ejecutando. Al hacer esta manipulación en un dispositivo, estamos naturalmente alterando el estado inicial y esto nos puede llevar a tener problemas. Cierto es, que en la mayoría de los casos el proceso es reversible y se pueden restablecer los valores iniciales y en el peor de los casos volver a instalar la ROM oficial.

Android ofrece una amplia gama de aplicaciones para todo tipo de ocasiones y propósitos, pero cuando queremos hacer algo un poco "más avanzado" siempre se requieren permisos root. Con el acceso root podemos usar programas de desinstalación de bloatware (apps preinstaladas), además de Firewall, sistemas de gestión para el control multi-touch con gestos y todas aquellas aplicaciones para root que modifican los archivos del sistema.

Por otra parte hay aplicaciones que, aunque funcionan sin permisos root, ofrecen muchísimas más funciones cuando sí los tenemos. Con permisos de root se pueden modificar los archivos de sistema a gusto personal, incluyendo sonidos de sistema y animación de arranque.

En entornos Android tenemos diferentes tipos de rooteo, y no todos tienen las mismas implicaciones desde el punto de vista forense, por lo que debemos evaluar en cada caso cuál necesitamos y actuar en consecuencia.

- Root permanente: Es el más habitual. Entramos al dispositivo y somos root. Normalmente es así porque se ha habilitado alguna ROM personalizada o se le han proporcionado al dispositivo binarios ARM para comandos como "su".

- Root temporal: Se elevan privilegios solo hasta el reinicio, normalmente tras aprovechar una vulnerabilidad en ejecución. Esta forma es ideal desde el punto de vista práctico, si bien entraña riesgos porque se va a modificar en dinámico, puede causar inestabilidad y aprovechar vulnerabilidades para elevar privilegios puede ser muy discutido desde el punto de vista del proceso.
- Root recovery: En este modo se habilita una partición de recuperación en el dispositivo que permite el acceso privilegiado solo cuando se carga dicha partición. Este tipo de Rooteo es el más indicado para procedimientos forenses. La problemática es que cada fabricante maneja las particiones de recuperación de una manera diferente y en algunos casos no será trivial.

Herramientas de Rooteo

A la hora de enfrentarse a un proceso de estas características pueden surgir una serie de complicaciones por lo que es importante tener en cuenta las dificultades inherentes al análisis de dispositivos móviles. La principal dificultad a la que nos enfrentamos es a la multitud de modelos distintos que existen en el mercado. Existe un informe de la empresa OpenSignal, donde esta certifica la existencia de más de 24.000 modelos distintos de terminales que incluyen sistema operativo Android. Esto dificulta significativamente el proyecto que estamos realizando y es por eso que no podremos tratar todas las técnicas o procedimientos.

Como es necesario conocer un amplio abanico de métodos, técnicas y herramientas así como los criterios necesarios para poder evaluar la idoneidad de utilización de unas respecto a otras, a continuación solo presentaremos las más comunes.

- Towelroot: Esta herramienta no necesita de un Pc para realizar el proceso de Rooteado. Utiliza la vulnerabilidad CVE-2014-3153. Se realiza mediante la instalación y ejecución de una app en el propio dispositivo, por tanto es necesario tener acceso a él y conocer el código de acceso. Su creador asegura que este exploit lograr conseguir privilegios de Root en todos los kernels compilados antes de 3 de Junio del 2014, eso indica con la ultima versión de Android 4.4.2 KitKat.
- Kingo Root: Es una herramienta de Rooteo muy conocida que tiene versión para ejecutarla desde un Pc y versión para ejecutarla desde un dispositivo Android. Para utilizar la versión de Pc es necesario tener habilitado el modo depuración USB del terminal. Es compatible con la gran mayoría de dispositivos del mercado y sus diferentes versiones de Android.
- SRSRoot: Es un software para Pc que realiza el Rooteo de un dispositivo Android de forma sencilla. El método es similar al de KingoRoot, es necesario conectar el dispositivo con el cable de datos al Pc, y además tener el “Modo depuración USB” activado. También es necesario disponer de los drives para el dispositivo en cuestión sea reconocido por el Pc. Soporta varias versiones de Android, desde la 1.5 hasta la 6.0.1, y dispone de una opción para desrootear el dispositivo.
- VRoot: Actualmente ha cambiado el nombre y ahora se llama iRoot. Ésta es una de las herramientas genéricas que más terminales Rootean, realizándose todo el proceso casi automáticamente. Como Kingo Root, es válida para Windows y también dispones de un app para Android.

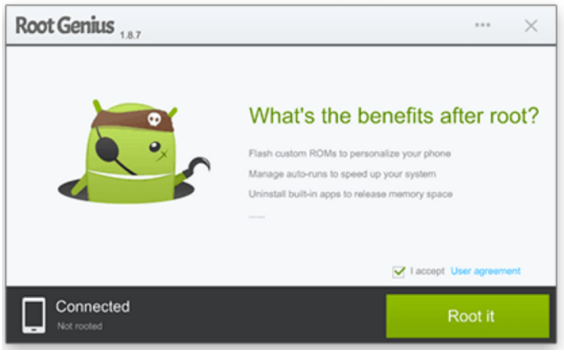
- **Root Genius:** Herramienta para realizar root en Android que precisa ser instalada en un Pc. Ha sido creada en China y soporta más de 10.000 modelos smartphones con Android. Se caracteriza por ser muy sencilla de utilizar, con solo un clic realiza el proceso. Es necesario tener conectado el terminal al Pc mediante cable y el modo depuración USB activado. Compatible con Android 2.2 a la 5.0.2.
- **FramaRoot:** Esta aplicación no requiere un ordenador. No funciona en tantos dispositivos ni marcas como las herramientas anteriores, pero también es una opción a tener en cuenta. Para su ejecución basta con instalar la app, ejecutarla y seguir el asistente.

Como Rootear un dispositivo

En el apartado anterior hacíamos una breve descripción de algunas de las diferentes herramientas que existen actualmente para realizar el proceso de Rooteo en un dispositivo Android. Cierto es que existen diferentes métodos para realizar el procedimiento, dependiendo de la versión de Android, del tipo de dispositivo y de la herramienta necesaria para realizar el proceso. Aunque ninguno de ellos requiere de una capacidad o conocimientos técnicos amplios, en este apartado explicaremos como realizarlo utilizando algunas de las herramientas antes descritas.

Root Genius

Se trata de una herramienta de Rooteo con un solo clic (one click root) y requiere de un PC con Windows para ser ejecutada. El exploit utilizado para Rootear su dispositivo Android puede ser detectado como malware por algún tipo de software antivirus.

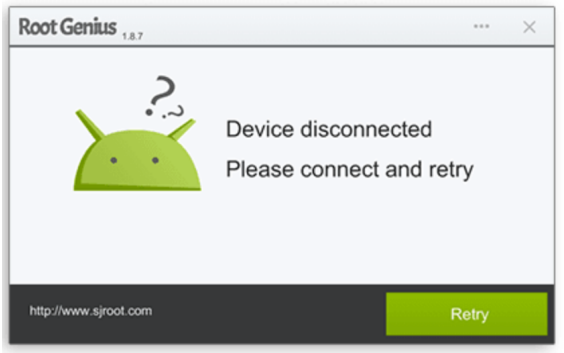


Captura 1 de la aplicación Root Genius.

Los pasos a seguir son los siguientes:

1) Activa el modo de depuración USB en el dispositivo Android. Ir a Ajustes -> *Opciones de desarrollo -> Depuración USB, o dependiendo de la versión de Android puede estar en el siguiente menú: Ajustes -> Aplicaciones -> Desarrollo -> Depuración USB.

**Las opciones de desarrollo pueden estar ocultas en terminal Android. Si quieres mostrarlas debes ir a Ajustes -> Acerca del teléfono -> Número de compilación y hacer clic sobre esta opción entre 5 y 8 veces.*



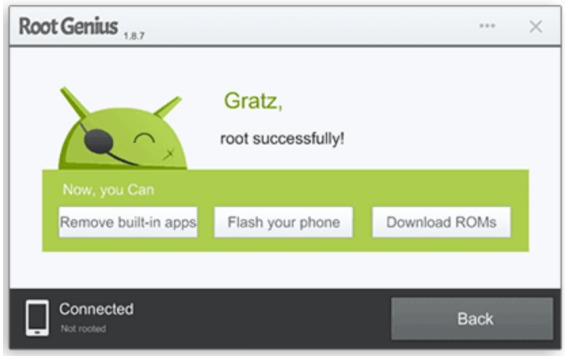
Captura 2 de la aplicación Root Genius.

2) Asegurarse de tener instalados los drivers USB apropiados para el dispositivo en el PC.

3) Conectar el dispositivo Android al PC. Root Genius lo debe detectar correctamente. Si no es así alguno de los pasos anteriores no se habrá realizado correctamente. Deberemos comprobar que los drivers USB están bien

instalados y que el modo de depuración USB esta activado en el terminal Android. Si todo ha ido bien deberás ver algo como la captura que aparece en la parte derecha.

4) Clic en el botón "Root it" para hacer el Root al dispositivo Android. Root Genius rooteará automáticamente el terminal Android. Una vez que el proceso de Root finalice el dispositivo Android estará rooteado correctamente. Si todo ha ido bien, deberás ver una pantalla como la que vemos a la derecha.



Captura 3 de la aplicación Root Genius.

Para comprobar si el dispositivo ha sido Rooteado correctamente, podemos comprobarlo de diferentes formas. La más habitual es instalando una aplicación para realizar la comprobación, una de las más conocidas es Root Checker.

Kingo Root

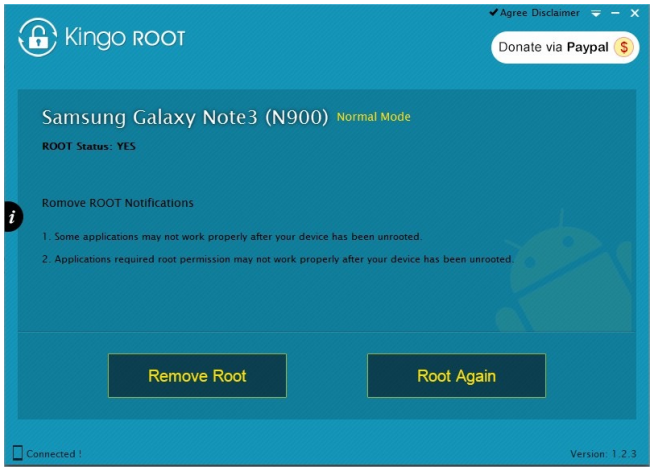
Esta aplicación, tanto en PC como en la versión de la APK, ofrece un acceso sencillo y rápido para a rootear Android. A continuación haremos una explicación breve de como utilizar esta herramienta:

Una vez que hemos instalado correctamente la aplicación debemos realizar una serie de pasos comunes a todos los terminales:

Iniciar la aplicación y conectar el terminal con el modo depuración USB activado, este modo se activa en las opciones de desarrollador, que por lo general vienen ocultas y debemos desbloquearlas pulsando 7 veces sobre el texto Número de Compilación que encontramos en Ajustes > Acerca del dispositivo. Tener correctamente instalados los drivers del dispositivo para que el equipo automáticamente instale el terminal en modo depuración USB.

Tras esto, la herramienta reconocerá el dispositivo que has instalado y si cumples los requisitos activará un botón naranja que dice ROOT, pulsamos este botón y comenzará el método.

Si por algún motivo necesitases realizar alguna acción durante el proceso, la aplicación te informa perfectamente y paso a paso de lo que debemos hacer. Al acabar el proceso ya tendremos rooteado nuestro terminal.



Captura de la aplicación Kingo Root.

FramaRoot

Esta herramienta se caracteriza por no necesitar de un Pc para realizar el proceso de Rooteo. Se instala la app en forma de apk directamente en el terminal, y desde esta se realiza todo el proceso.

1) El proceso de instalación es fácil y sencillo, simplemente tienes que descargar framaroot a través de su página web http://framaroot.net/.

2) Luego proceden a instalarla en su dispositivo, para ello deben de tener habilitada la opción de "Fuentes desconocidas" desde "Ajustes -> Seguridad" y marcar la casilla de "Fuentes desconocidas".

3) Al ejecutar Framaroot podemos seleccionar 3 opciones:

- Install Superuser.
- Install SuperSU.
- Unroot (Si quieres desrootear su dispositivo).

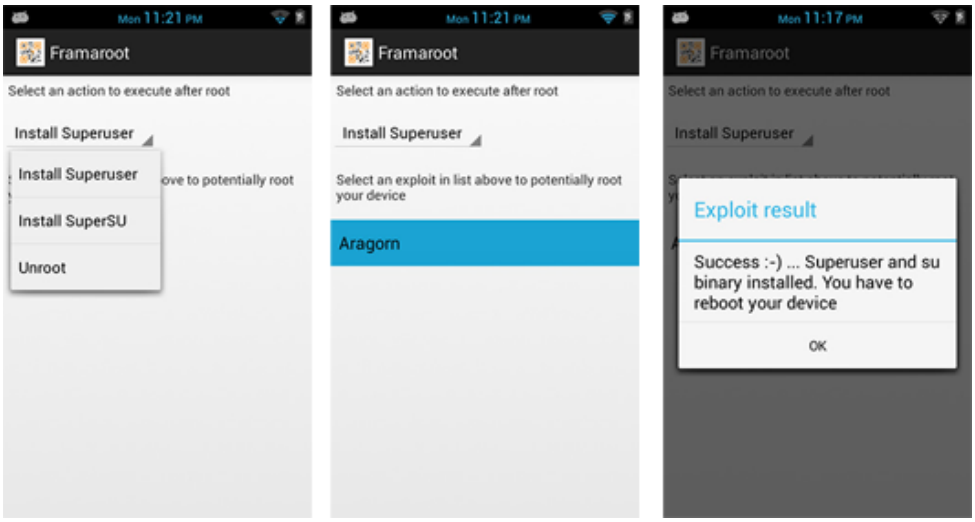
4) Seleccionar Install Superuser.

5) En la parte de inferior aparecerá una lista de 3 exploits, habrá que probar hasta encontrar el que es compatible con el dispositivo.

6) Si no es compatible nos dará un mensaje de error: "Failed :- (Try another expolit if available"

Si es compatible y si realiza el root saldrá este mensaje: "Success : –) Superuser and su binary installed. You have to reboot your device".

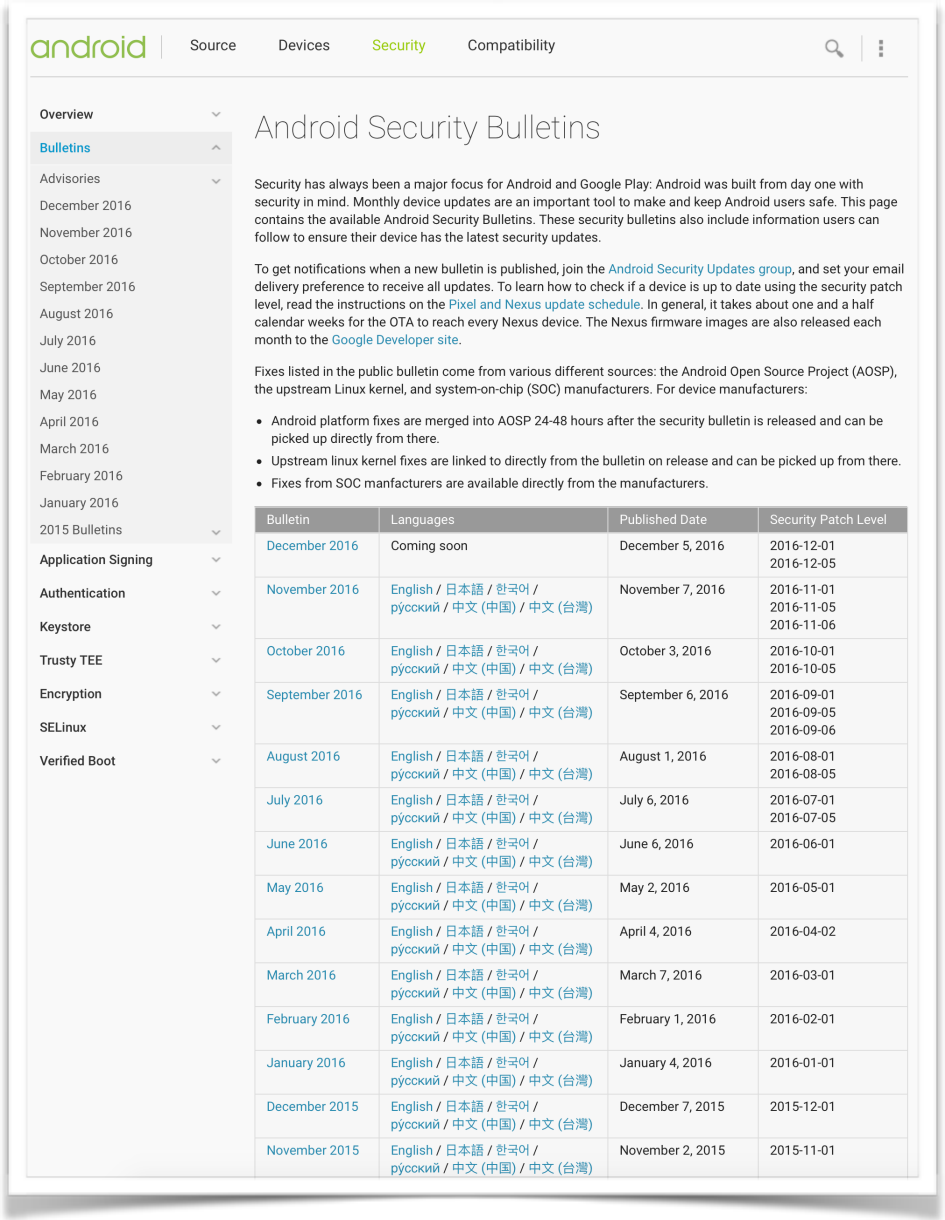
7) Solo habrá que reiniciar el dispositivo y ya tendremos rooteado, de una manera fácil, el terminal.



Captura de la aplicación FramaRoot.

Vulnerabilidades públicas en Android

Desde la primera versión del sistema operativo Android, han ido apareciendo problemas de seguridad. Estos se han ido solucionando en las siguientes versiones publicadas, aunque no siempre todos los dispositivos puedan acceder a ella. Por eso, en esta parte del proyecto vamos a conocer el nivel de seguridad y las posibles amenazas a las que está expuesto el dispositivo a analizar y qué vulnerabilidades hay publicadas que afecten al terminal.



Listado de los boletines de seguridad publicados en android.com

En la página oficial de Android podemos encontrar los boletines de seguridad que publica Google, en los que aparecen las vulnerabilidades que afectan a su sistema operativo, con su correspondiente CVE y la gravedad

de esta. También anuncia que los parches para estas vulnerabilidades están disponibles en el repositorio de Android Open Source Project (AOSP). La web es https://source.android.com/security/bulletin/.

Security vulnerability summary

The tables below contains a list of security vulnerabilities, the Common Vulnerability and Exposures ID (CVE), the assessed severity, and whether or not Google devices are affected. The severity assessment is based on the effect that exploiting the vulnerability would possibly have on an affected device, assuming the platform and service mitigations are disabled for development purposes or if successfully bypassed.

2016-11-01 security patch level—Vulnerability summary

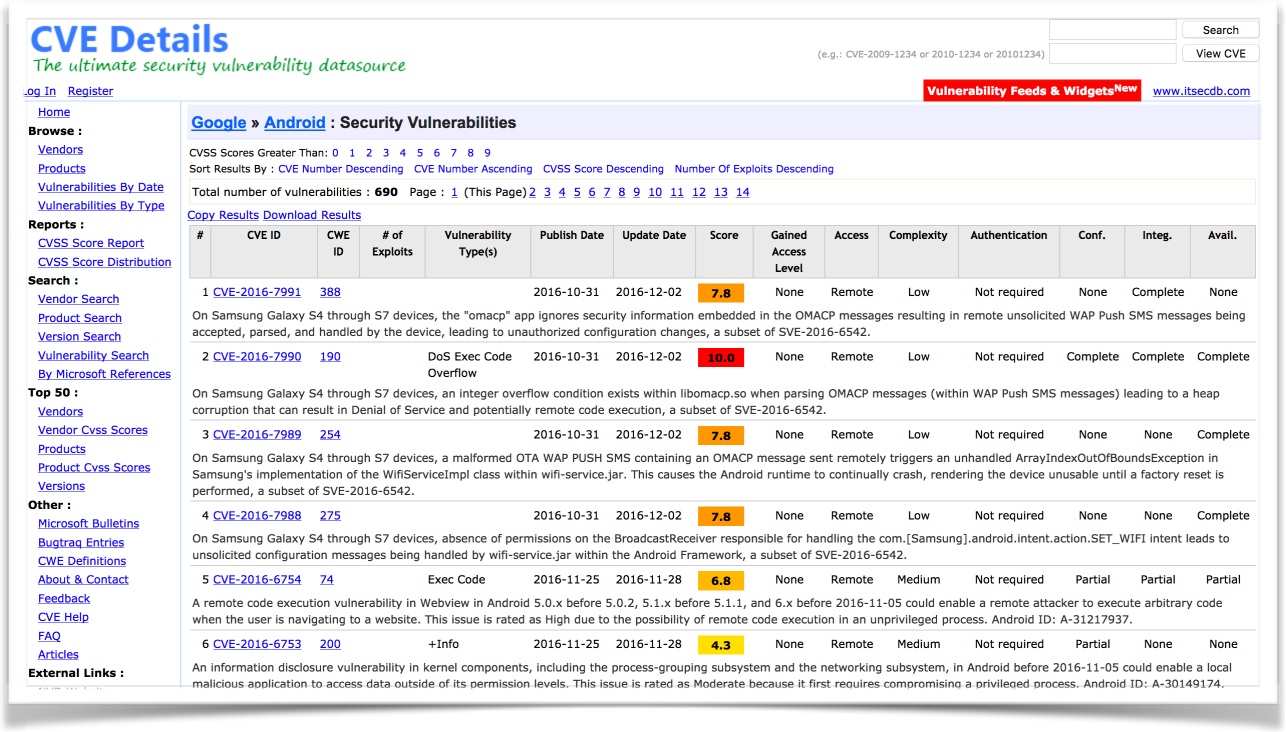
Security patch levels of 2016-11-01 or later must address the following issues.

Issue	CVE	Severity	Affects Google devices?
Remote code execution vulnerability in Mediaserver	CVE-2016-6699	Critical	Yes
Elevation of privilege vulnerability in libzipfile	CVE-2016-6700	Critical	No*
Remote code execution vulnerability in Skia	CVE-2016-6701	High	Yes
Remote code execution vulnerability in libjpeg	CVE-2016-6702	High	No*
Remote code execution vulnerability in Android runtime	CVE-2016-6703	High	No*
Elevation of privilege vulnerability in Mediaserver	CVE-2016-6704, CVE-2016-6705, CVE-2016-6706	High	Yes
Elevation of privilege vulnerability in System Server	CVE-2016-6707	High	Yes
Elevation of privilege vulnerability in System UI	CVE-2016-6708	High	Yes
Information disclosure vulnerability in Conscrypt	CVE-2016-6709	High	Yes
Information disclosure vulnerability in download manager	CVE-2016-6710	High	Yes
Denial of service vulnerability in Bluetooth	CVE-2014-9908	High	No*
Denial of service vulnerability in OpenJDK	CVE-2015-0410	High	Yes
Denial of service vulnerability in Mediaserver	CVE-2016-6711, CVE-2016-6712, CVE-2016-6713, CVE-2016-6714	High	Yes

Detalle del boletín de seguridad publicados el 01/11/2016 en android.com

En la web de CVE Details también existe un repositorio, donde aparecen todas las vulnerabilidades reportadas desde los inicios de Android, que ya suman un total de 690.

La web es la siguiente: https://www.cvedetails.com/vulnerability-list/vendor_id-1224/product_id-19997/Google-Android.html



Listado de los vulnerabilidades públicas en la web de https://www.cvedetails.com

En la imagen superior se puede observar la captura de la web CVE Details, donde aparecen todas las vulnerabilidades registradas e información relacionada con su catalogación. Dentro de cada una aparece se puede observar una descripción del fallo, el Score, tipo de vulnerabilidad, las versiones y productos a los que afecta, etc, incluso si existe algún expolio público para esta.

Ya que estamos hablando sobre las vulnerabilidades en Android, vamos a comentar una de las más importantes y conocidas hasta la fecha. Hace un tiempo se encontraron un total de 4 vulnerabilidades de seguridad en los dispositivos que equipaban un chip de Qualcomm, sin importar la versión de Android que estaban usando. A esta vulnerabilidad se le puso el nombre de Quadrooter.

Quadrooter es una vulnerabilidad que se considera de alto riesgo por aprovecharse de vulnerabilidades en el escalado de privilegios. Además, según la empresa de seguridad Check Point, afecta a casi mil millones de dispositivos alrededor del mundo. Un atacante que se aproveche de Quadrooter solo tendría que engañar al usuario para que instale una app maliciosa que no pide permisos especiales. Si el atacante consigue aprovecharse de alguno de los fallos, podrá obtener acceso root y controlar el dispositivo atacado.

Si bien la mayoría de errores de esta vulnerabilidad ya han sido reparados en los parches más recientes, existen dispositivos que no han podido actualizarse. La información que publican desde Check Point es la siguiente: *"Dado que los drivers vulnerables están pre-instalados en los dispositivos, sólo pueden ser corregidos mediante una actualización por parte del distribuidor. Para que el distribuidor pueda enviar la actualización, primero deben recibir los drivers corregidos por parte de Qualcomm."*

Se recomienda no instalar aplicaciones que desconozcamos la procedencia y hacerlo solo desde Google Play, pero aún así no estaremos 100% seguros.

Métodos de explotación y acceso

Acceso al dispositivo

En esta sección vamos a explicar las diferentes formas que existen y que podría utilizar un atacante para acceder a un dispositivo Android. Explicaremos de forma breve las técnicas para evadir el código de bloqueo, como saber si tenemos acceso Root en un dispositivo, en que consiste en modo depuración USB, como acceder mediante ADB, etc.

En primer lugar explicaremos qué metodologías y fases del proceso nos podemos encontrar a la hora de intentar adquirir los datos de un dispositivo móvil con Android. Para ello es necesario conocer un amplio abanico de métodos, técnicas y herramientas así como los criterios necesarios para poder evaluar la idoneidad de utilización de unas respecto a otras.

A grandes rasgos podemos describir 3 métodos distintos a la hora de extraer datos de un terminal:

- Adquisición física: Consiste en realizar una réplica idéntica de los datos del dispositivo original. La ventaja de este procedimiento es que permite buscar elementos eliminados. Su desventaja es su complejidad respecto a los otros métodos y el tiempo que de realización.
- Adquisición lógica: Este proceso consiste en realizar una copia de los datos almacenados en el dispositivo de forma lógica. Es decir, se utilizan los mecanismos implementados por el fabricante, aquellos que son utilizados para sincronizar el terminal con un ordenador. Es un proceso mucho más sencillo que el anterior, pero no permite acceder a toda la información.
- Adquisición del sistema de ficheros: permite obtener todos los ficheros visibles mediante el sistema de ficheros, pero esto no incluye ficheros eliminados o particiones ocultas. Si nos puede resultar suficiente utilizar este método, esto supondría una complejidad menor que la adquisición física. Para llevarlo a cabo se utilizan los mecanismos integrados en el sistema operativo para el copiado de ficheros, como es el Android Device Bridge (ADB) en este caso. Mediante este método es posible recuperar cierta información eliminada porque en algunos casos utilizan bases de datos SQLite para almacenar gran parte de la información.

A la hora de seleccionar el método más adecuado, tenemos que tener en cuenta multitud de aspectos como el tiempo del que disponemos, que información queremos extraer, el tipo de acceso que tenemos al

dispositivo, y en concreto, cuál es la finalidad de acceder a al dispositivo, conseguir algún dato, tomar el control, realizar una auditoría o un análisis forense, etc.

Por tanto, para saber como deberíamos actuar en el momento de elegir el método a seguir, podemos hacer uso del siguiente diagrama, que nos ayudará a tomar esa decisión.

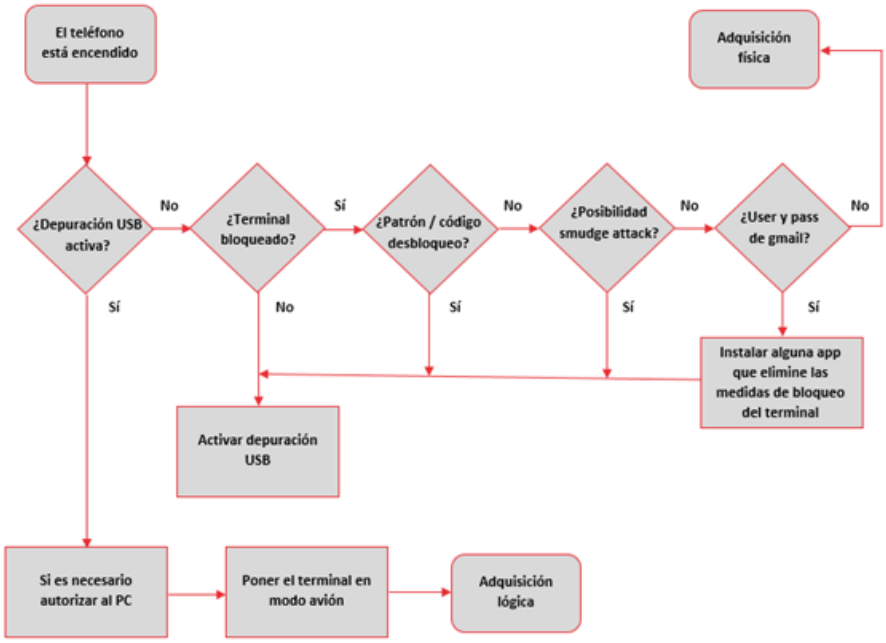


Diagrama con los posibles escenarios y los pasos a seguir. Fuente: https://www.certsi.es

Para aclarar los posibles casos que se nos pueden dar, a continuación describiremos alguna de los procesos mencionados en el diagrama.

En primer lugar explicaremos qué es el modo “depuración USB” y su utilidad. El modo “Depuración USB” (USB Debugging) está pensado principalmente para desarrolladores. Este modo abre el acceso directo al sistema para el SDK de Android (Software Development Kit) y nos sirve para conectar y pasar información entre el smartphone y el ordenador. Las herramientas que utilizamos para llevar esta tarea acabo son el ADB (Android Debug Bridge) para la mayoría de los smartphone o algunos otro programas según el fabricante como por ejemplo Odin para Samsung.

Así que el primer paso para cambiar alguna parte del software de un smartphone con Android es activando la depuración por USB. Esto nos abre la puerta del Root, instalar un custom recovery, instalar una nueva ROM o volver a reinstalar por completo es sistema operativo de fábrica.

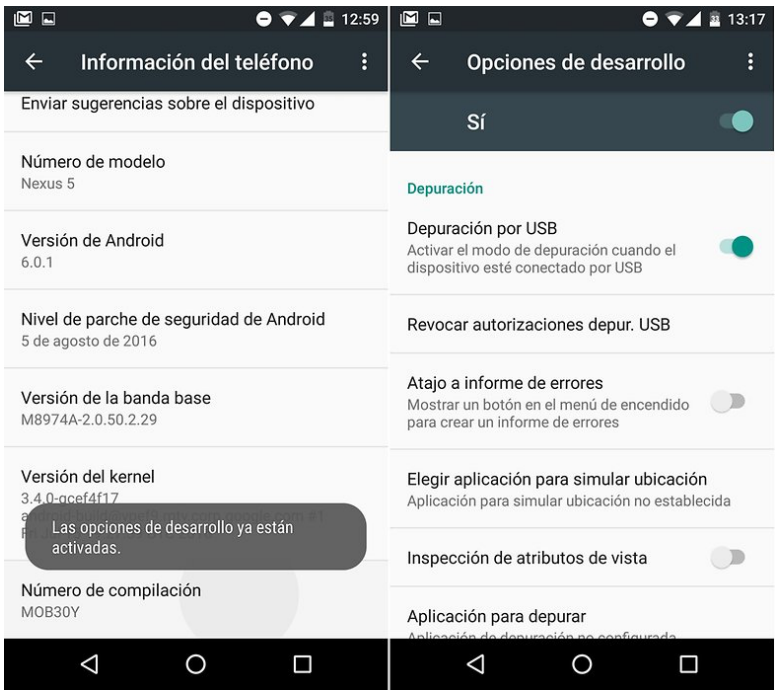
Por supuesto si queremos desarrollar una aplicación para Android, activar la depuración por USB es la piedra angular que nos permitirá probar la aplicación en el dispositivo mientras la estamos desarrollando. De esta manera podemos ver inmediatamente que efectos tienen los cambios que introducimos en el código de nuestra aplicación.

No hay necesidad de tener la depuración por USB constantemente activa en el dispositivo ya que esto supone un riesgo para la seguridad del dispositivo. En el caso de robo si la depuración por USB está activa será más fácil acceder a los datos que contiene el terminal o a instalar otro sistema operativo.

Esta opción se encuentra en el menú oculto para desarrolladores. Para activar este menú debemos entrar en "Ajustes -> Información del teléfono" y pulsar varias veces sobre el "Número de compilación" hasta que aparezca el aviso de que las opciones de desarrollo están activas. Ahora en el menú de "Ajustes" vamos a tener otro apartado llamado "Opciones de desarrollo". Si bajamos un poco dentro de este nuevo menú vamos a encontrar un apartado llamado "Depuración" y su la primera opción es "Depuración por USB".

Los pasos a seguir en Android 4.0 y 4.1 son más sencillos aún. Sólo tendremos que irnos a Ajustes, una vez dentro buscamos Opciones para desarrolladores que nos aparecerá sin necesidad de activarla y marcamos en Depuración USB.

Para poder lograr la comunicación entre el ordenador y el dispositivo Android tendremos que autorizar esta conexión. Por tanto conectaremos el dispositivo a nuestro ordenador y en nuestro Android nos aparecerá un mensaje donde nos pregunta si queremos permitir la depuración USB. Marcamos que lo permita para este ordenador.



Captura de pantalla en Android donde activar la "Depuración por USB".

Una vez conectados mediante USB al equipo y con el smartphone con el modo depuración USB activado, tenemos varias herramientas para acceder. De estas opciones veremos las 2 más conocidas, que son el ADB y Fasboot.

Las siglas ADB significan Android Debug Bridge y se corresponden con una herramienta de software que nos permite interactuar con nuestro smartphone Android desde un ordenador. El ADB es una parte fundamental de Android Studio, el software para desarrollar aplicaciones en Android. Así, a través de ADB podemos ejecutar comandos para copiar archivos desde el ordenador al teléfono, del teléfono al ordenador o reiniciar el dispositivo en el modo bootloader.

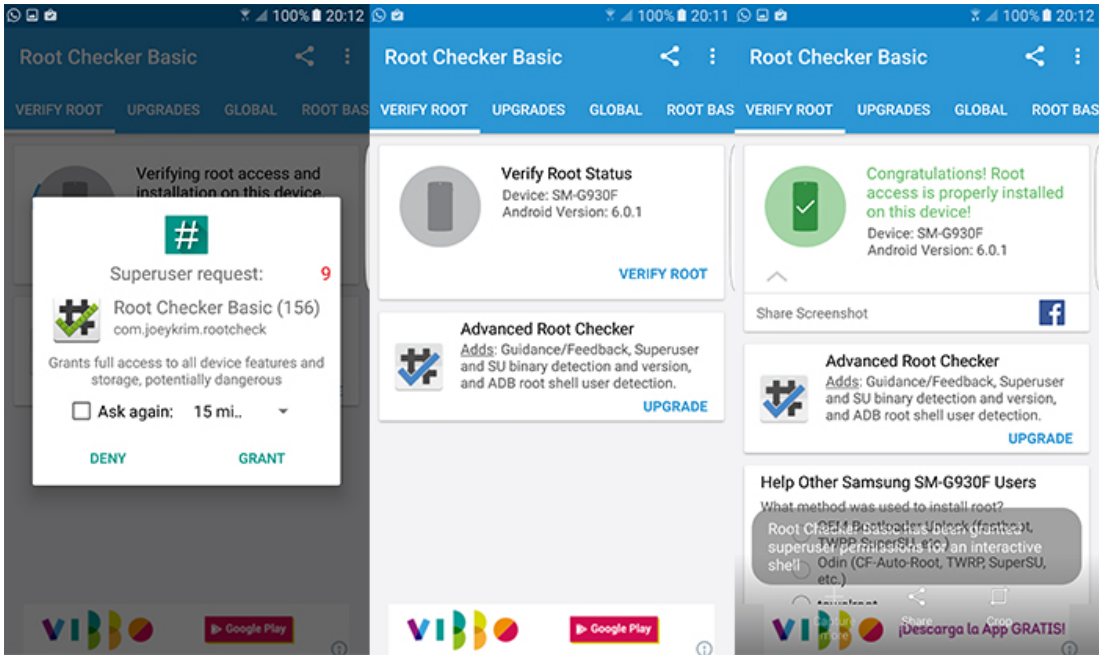
El Fastboot también es una herramienta de software con la que podemos comunicarnos y modificar partes de un smartphone Android, previamente conectado a través de un cable USB, desde un ordenador. Con el Fastboot vamos a poder desbloquear el bootloader, flashear un recovery, flashear el firmware completo o reiniciar el dispositivo en modo recovery. Básicamente, desde un ordenador: con en el ADB podemos comunicarnos con un smartphone Android que está encendido y su sistema Android funcionando, con el

Fastboot podemos comunicarnos con el smartphone Android cuando lo hemos arrancado en modo bootloader. Con estas dos herramientas vamos a poder cambiar el software de nuestro smartphone o por lo menos acceder a él, eso sí, siempre a través de un cable USB como hemos comentado antes.

Llegado a este punto debemos tener en cuenta varias cosas, en primer lugar debemos saber si tenemos acceso Root al terminal Android y conocer el código de desbloqueo del dispositivo, en caso contrario debemos saber que técnicas podemos utilizar para evadir el bloqueo de pantalla.

Existen varios métodos para saber si tenemos permisos de superusuario en el dispositivo. Los 3 pasan por instalar una app, y verificarlo desde esta.

La primera que veremos es Root Checker, la podemos encontrar en Google Play y tiene la única función de verificar si el dispositivo tiene acceso Root o no. Root Checker tiene un botón que al ser pulsado procederá a mostrar el estado en que se encuentra el terminal.



Captura de la app Root Checker.

En segundo lugar podemos utilizar algo más sofisticado. Este método se basa en comprobar desde el terminal de Android, mediante un comando, si el smartphone está Rooteado. Si no tenemos instalada la aplicación deberemos instalarla desde Google Play, y la encontraremos con el nombre de Terminal Emulator. El comando a utilizar es simplemente “su” y el resultado será una ventana emergente pidiéndonos permisos de root para la aplicación, en caso de no estar Rooteado, la app no mostrará nada.

En tercer y último lugar comentaremos la aplicación SuperSu para conocer el estado del terminal. SuperSu es una aplicación que controla los permisos de superusuario de otras aplicaciones. Es, por lo tanto, un imprescindible si queremos ser root y en consecuencia un método para saber si lo somos. Normalmente, tras realizarle root a cualquier dispositivo Android, se instala por defecto. Por lo tanto, bastará con abrirla, y si todo

ha ido bien, nos informará de las apps que nos han pedido este acceso y si se lo hemos concedido o no. Por el contrario, si no somos superusuarios, nos informará de ello y puede que nos ofrezca serlo mediante el flasheo de un archivo .ZIP en nuestro recovery.

Respecto al problema de encontrarnos con un terminal que desconozcamos la forma de desbloquearlo, porque tenga un sistema de seguridad activado, vamos a explicar diferentes técnicas para evadir estas protecciones.

- Smudge Attacks. Esta técnica se basa en analizar la grasa corporal que dejan los dedos en la pantalla al desbloquear el dispositivo, y por medio de una fotografía de calidad averiguar el código de desbloqueo mediante las marcas que quedaron en la pantalla del dispositivo. En el siguiente papper detallan esta aproximación desde una manera formal: http://static.usenix.org/event/woot10/tech/full_papers/Aviv.pdf.
- Pattern Lock Crack: Evadir esta protección es muy sencilla solo si se tiene acceso al modo debug del dispositivo rooteado. El patrón se almacena en el dispositivo en forma de hash en el fichero /data/system/gesture.key. Si tenemos acceso mediante ADB como root, podemos eliminar el fichero y crear uno nuevo vacío mediante el comando: *"rm gesture.key touch gesture.key".* A partir de ese momento, cualquier gesto desbloqueará el dispositivo. Si no tenemos acceso al modo debug, es posible a través de la interfaz JTAG, pero no entraremos en detalles por no extendernos más, el proceso está descrito en el siguiente enlace: http://forensics.spreitzenbarth.de/2012/02/28/cracking-the-pattern-lock-on-android/
- Password Crack y PIN Crack: Evadir estas protecciones es muy sencillo solo si se tiene acceso al modo debug del dispositivo rooteado. La contraseña y/o PIN se almacena en el dispositivo en forma de hash en el fichero /data/system/password.key. Si tenemos acceso ADB de root, podemos eliminar el fichero y crear uno nuevo vacío con: *"rm password.key touch password.key".* A partir de ese momento, cualquier contraseña y/o PIN desbloqueará el dispositivo.

Configuración del dispositivo y recomendaciones

En este apartado vamos a realizar un análisis de los controles de acceso y privacidad del sistema operativo Android, de los servicios activos innecesarios y el tratamiento de los errores de configuración.

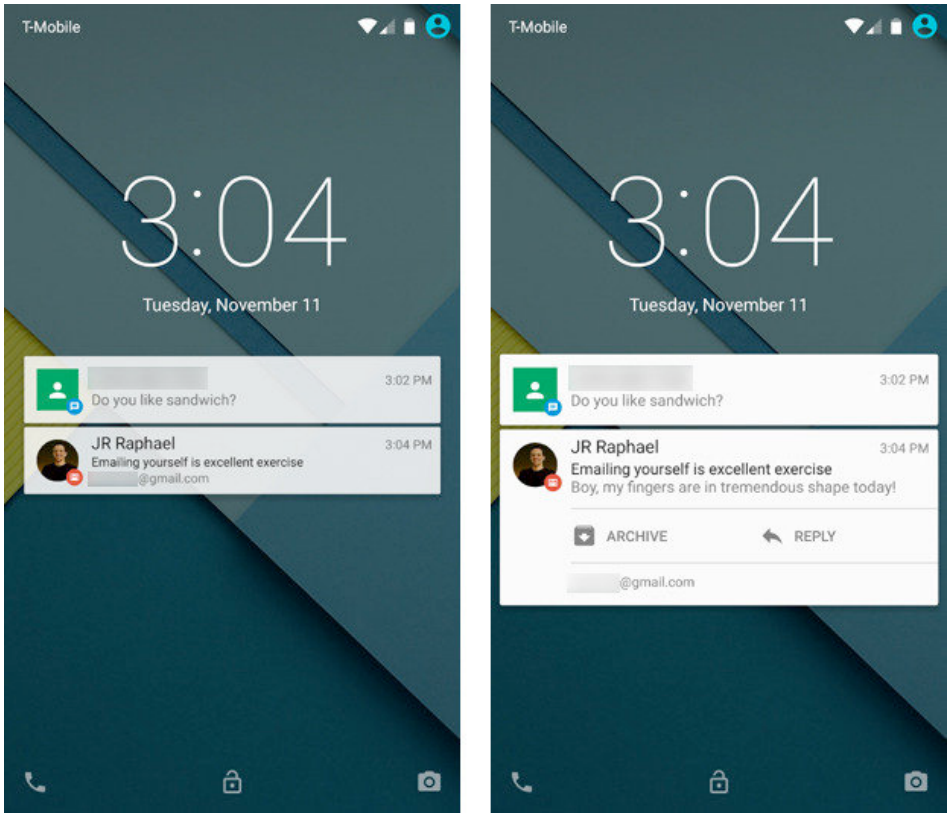
En primer lugar hablaremos de las actualizaciones, ya que son fundamentales porque en algunas ocasiones van a definir si el equipo es vulnerable o no. A través de estas, se corrigen errores, que podrían ser fallas de seguridad mediante las cuales un atacante podría ingresar al sistema, vulnerarlo y robar información. Por tanto debemos tener el dispositivo actualizado a la última versión posible y en caso de no poder debido a limitación del fabricante del dispositivo o similar, intentar averiguar si la versión de Android que tenemos instalada es una versión segura.

Como hemos comentado anteriormente, la posibilidad de disponer de acceso físico al dispositivo móvil nos abriría muchas puertas a la hora de acceder a los datos del terminal. Por este motivo se debe activar los sistemas de seguridad de bloqueo, ya que la ausencia de estos permitiría a un potencial atacante acceder a los contenidos del mismo o hacer uso de los servicios de telefonía móvil disponibles. Por defecto, los dispositivos móviles basados en Android no disponen de un PIN o contraseña de acceso para bloquear

accesos no autorizados al terminal. Anteriormente ya comentamos los diferentes sistemas que ofrece Android para proteger el dispositivo.

Desde el punto de vista de la seguridad se recomienda no habilitar el sistema de desbloqueo mediante patrón. Como hemos visto anteriormente, este sistema es vulnerable a la técnica Smudge Attack. Se recomienda el uso de protección mediante contraseña alfanumérica y en segundo lugar de código número de 6 o más cifras.

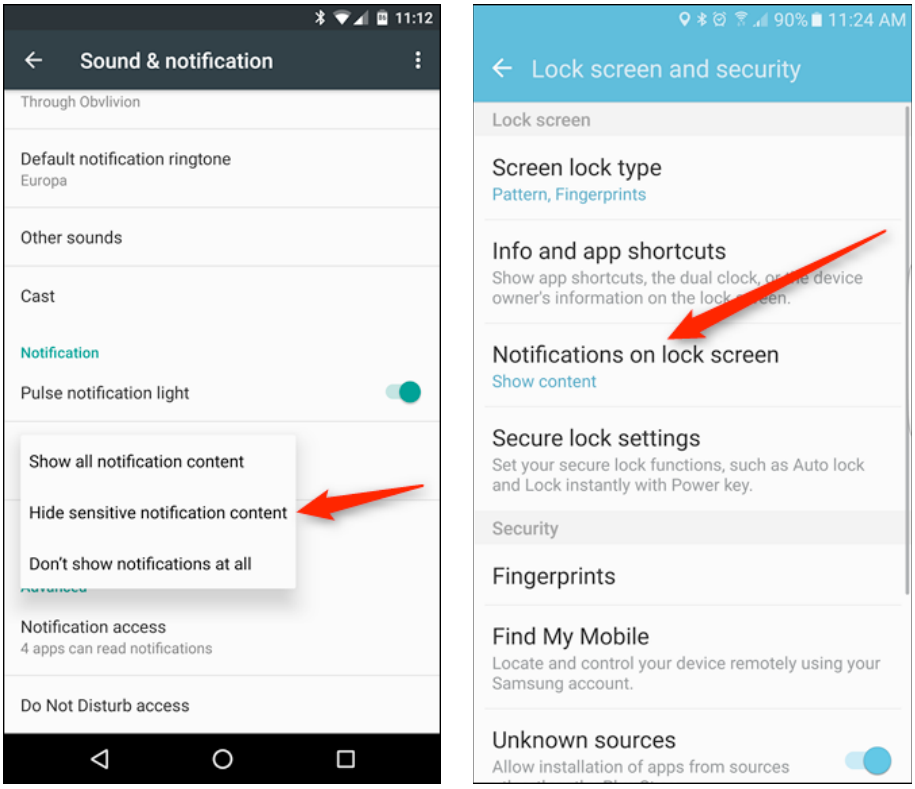
La pantalla de desbloqueo del terminal en Android permite, sin necesidad de introducir el PIN o contraseña de acceso y, por tanto, sin necesidad de desbloquear el teléfono, junto a la hora y la fecha, la previsualización del nombre del operador de telecomunicaciones y de la barra de estado, que incluye los iconos de conectividad y de notificaciones.



Captura de la pantalla bloqueada de un dispositivo Android.

La pantalla de bloqueo puede desvelar por tanto información potencialmente confidencial a un potencial atacante, pero sólo temporalmente. Por ejemplo, en el caso de recibirse un mensaje SMS, el comienzo del mismo es mostrado durante un segundo en la barra de estado. La información del SMS mostrada incluye el número de teléfono del remitente y los primeros caracteres del mensaje en la parte superior de la pantalla de desbloqueo. Lo mismo ocurre con otras aplicaciones de mensajería como Twitter, WhatsApp, etc.

Por tanto desde el punto de vista de seguridad, en el caso de gestionar información sensible y confidencial, se recomienda deshabilitar esta previsualización. Dependiendo de la versión de Android y del dispositivo, el menú



Captura de la pantalla del menú "Ajustes -> Notificaciones" de un dispositivo Android.

donde aparece la opción de deshabitar las notificaciones puede variar.

Respecto a la configuración por defecto del terminal Android, el dispositivo móvil está configurado por defecto con valores fijados por el fabricante del sistema operativo, el fabricante del hardware o el operador de telefonía móvil, que pueden desvelar detalles del terminal o de su propietario.

En caso de desvelarse el nombre del dispositivo en sus comunicaciones, como por ejemplo en el tráfico DHCP, éste podría revelar detalles tanto del fabricante del dispositivo móvil, como por ejemplo el modelo del terminal, como del propietario. Estos datos serían muy útiles para un potencial atacante interesado en explotar vulnerabilidades en ese tipo concreto de dispositivo móvil o interesado en realizar ataques dirigidos hacia una persona concreta. Para evitar la revelación de información sensible, se recomienda modificar esos valores existentes por defecto por valores elegidos por su propietario.

Las capacidades de localización del dispositivo móvil a través de GPS, o de las redes de datos (telefonía móvil y Wi-Fi), pueden ser activadas y desactivadas por el usuario en función de su utilización. El dispositivo GPS en Android está activo únicamente cuando se está haciendo uso de una aplicación con capacidades de localización, es decir, que hace uso del GPS. Existen estudios que reflejan que la información de localización intercambiada entre el dispositivo móvil Android y Google no es completamente anónima, ya que aparte de

incluir detalles de las señales inalámbricas recibidas y la ubicación del terminal, se incluye un identificador único del dispositivo móvil, afectando a la privacidad del usuario.

Se recomienda deshabilitar las capacidades de localización del dispositivo móvil salvo que se esté haciendo uso explícito de esta funcionalidad. En caso de habilitar dichas capacidades, se recomienda activar únicamente la localización mediante GPS, no haciendo uso de las capacidades de localización mediante redes Wi-Fi.

Servicios más avanzados, como Google Location History, disponible desde la versión 5.3 de Google Maps en Android permiten disponer de un histórico completo de la ubicación del usuario, incluso con análisis y porcentajes de dónde ha estado el usuario (casa, trabajo, viajando, etc). Pese a que se supone que este tipo de servicios avanzados son privados para el usuario, los mismos permiten disponer de un nivel demasiado detallado de información sobre la ubicación del usuario a lo largo del tiempo, y constituyen un riesgo relevante para la privacidad del mismo. En caso de haber habilitado este servicio es posible eliminar el histórico de ubicaciones previamente almacenado en Google. Se recomienda no hacer pública la ubicación en la que se encuentra el usuario en redes sociales o servicios avanzados similares con el objetivo de proteger su privacidad.

Android permite añadir información de la localización geográfica a las fotografías realizadas con la cámara del dispositivo móvil. Al realizar la fotografía se añaden las coordenadas GPS en el momento de tomar la instantánea a la cabecera EXIF de la imagen. Se recomienda no hacer uso de la funcionalidad que incluye la información del GPS en fotografías, especialmente para su publicación o distribución en Internet, salvo que se quiera hacer uso explícito de estas capacidades. En caso contrario, las imágenes revelarán los detalles exactos de dónde han sido tomadas.

Captura de pantalla en Android para activar almacenamiento USB.

El acceso al dispositivo móvil como unidad de almacenamiento USB puede estar disponible incluso con el modo de depuración habilitado en Android. En caso de conflictos y estar habilitado, basta con deshabilitar este modo (opción recomendada desde el punto de vista de seguridad), para disponer de las capacidades de almacenamiento USB únicamente. El modo de depuración se puede deshabilitar mediante el menú “Ajustes – Aplicaciones – Desarrollo”, desactivando la opción “Depuración USB”.

La principal recomendación de seguridad asociada a las comunicaciones Bluetooth (802.15) en dispositivos móviles es no activar el interfaz inalámbrico Bluetooth salvo en el caso en el que se esté haciendo uso del mismo, evitando así la posibilidad de ataques sobre el hardware del interfaz, el driver o la pila de comunicaciones Bluetooth, incluyendo los perfiles Bluetooth disponibles. El estado del interfaz Bluetooth puede ser verificado

de forma rápida y sencilla por el usuario del dispositivo móvil a través de la barra superior de estado.

La principal recomendación de seguridad asociada a las comunicaciones Wi-Fi (802.11) en dispositivos móviles es no activar el interfaz inalámbrico Wi-Fi salvo en el caso en el que se esté haciendo uso del mismo, evitando así la posibilidad de ataques sobre el hardware del interfaz, el driver o la pila de comunicaciones WiFi.

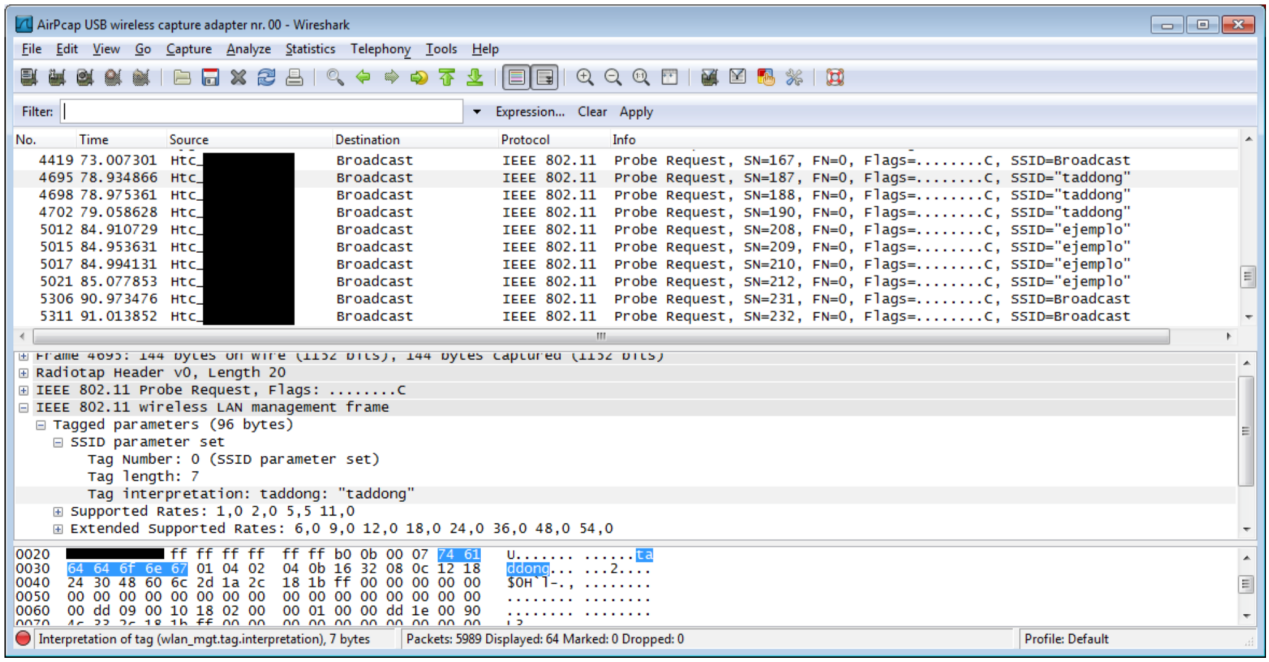
Android proporciona información en la barra superior de estado sobre la disponibilidad de redes Wi-Fi abiertas y si se está o no conectado a una red Wi-Fi actualmente. La notificación respecto a la disponibilidad de redes Wi-Fi abiertas puede ser configurada a través del menú “Ajustes –> Conexiones inalámbricas (y redes) –> Ajustes de Wi-Fi”, y en concreto mediante la opción “Notificación de red”.

Dado que desde el punto de vista de seguridad se desaconseja el establecer conexiones con redes Wi-Fi abiertas, se recomienda desactivar esta opción de notificación.

Captura de la pantalla de un dispositivo Android en el menú avanzado de Wi-Fi.

Con el objetivo de proteger el dispositivo móvil y aumentar su nivel de seguridad, se desaconseja el uso de redes inalámbricas públicas y abiertas, con un nivel de seguridad reducido o inexistente, sin mecanismos de autentificación y cifrado. Los riesgos de seguridad asociados a este tipo de entornos incluyen la captura e interceptación del tráfico y datos por parte de un potencial atacante, la posibilidad de inyección de tráfico para la realización de ataques directos y de accesos no autorizados contra el dispositivo móvil, ataques de suplantación de la red, y a ataques de denegación de servicio, entre otros.

Tampoco se recomienda añadir ninguna red Wi-Fi manualmente al dispositivo móvil, aunque es una práctica común desde el punto de vista de seguridad para disponer de mayor control sobre la configuración de la red a añadir. No se recomienda porque para todas las redes configuradas maualmente, Android generará tráfico que desvelará estas redes Wi-Fi disponibles en su lista de redes e intentará conectarse a ellas, escenario que



Captura del tráfico de red con Wireshark de un terminal Android, al conectarse a una red Wi-fi.

puede ser empleado por un potencial atacante para disponer de conectividad con el dispositivo móvil y proceder a explotar vulnerabilidades en el mismo.

Android dispone de capacidades nativas para actuar como punto de acceso Wi-Fi y router hacia Internet, proporcionando acceso a múltiples dispositivos. Debe tenerse en cuenta que, al igual que para el interfaz Wi-Fi y Bluetooth, para poder llevar a cabo la configuración del punto de acceso Wi-Fi a través del interfaz de usuario del dispositivo móvil es obligatorio activar esta funcionalidad por lo que se recomienda activar la funcionalidad y realizar las modificaciones de la configuración en un entorno seguro, para evitar potenciales ataques cuando los mecanismos de seguridad no han sido aún aplicados.

Se recomienda no habilitar las capacidades de punto de acceso Wi-Fi en el dispositivo móvil salvo que se quiera hacer un uso limitado y controlado de esta funcionalidad. En caso de ser utilizada, la red Wi-Fi asociada debe ser protegida y configurada mediante WPA2-PSK con una contraseña o clave precompartida suficientemente larga.

Las capacidades de mensajería de texto SMS de los dispositivos móviles permite el envío de mensajes cortos empleando la infraestructura GSM. El primer nivel de protección asociado al sistema de mensajería de texto pasa por la concienciación del usuario desde el punto de vista de seguridad, teniendo en cuenta que la implementación de este módulo en los dispositivos móviles actuales proporciona numerosas funcionalidades, y no únicamente la visualización de mensajes de texto. Entre las funcionalidades avanzadas se encuentra la posibilidad de mostrar contenidos de texto, contenidos web (HTML, JavaScript, etc.), gestionar datos binarios, como tonos de llamada, e intercambiar ficheros multimedia (audio, vídeo e imágenes). Debido a las diferentes vulnerabilidades asociadas al módulo de mensajería de texto en Android, se recomienda que el

usuario actúe con prudencia ante la lectura de nuevos mensajes de texto, especialmente los recibidos de fuentes desconocidas. Durante el año 2009 se publicaron diferentes vulnerabilidades de denegación de servicio (DoS) en el módulo SMS de dispositivos Android.

Para terminar este apartado, con el objetivo de disponer del mayor nivel de seguridad posible en el dispositivo Android y de manera general, se recomienda disponer siempre de la última versión de todas las aplicaciones instaladas, independientemente del desarrollador. La última versión debería solucionar todas las vulnerabilidades de seguridad públicamente conocidas.

Adicionalmente existen numerosas aplicaciones cliente o suites de software comercial disponibles a través de Google Play o de compañías comerciales cuyo objetivo es aumentar el nivel de seguridad del dispositivo móvil Android, como suites o soluciones de seguridad con antivirus, copias de seguridad, bloqueo remoto del terminal, borrado de datos y localización del terminal.

Malware en Android

La falta de conocimiento acerca de las amenazas para dispositivos móviles, expone a los usuarios a la pérdida de su información o la infección de su smartphone.

Muchos usuarios desconocen que las plataformas móviles son utilizadas por los desarrolladores de códigos maliciosos para enviar enlaces dañinos que redirigen al usuario a la descarga de malware, y es debido a ello que caen víctimas de los engaños.

Las amenazas existentes para las distintas plataformas móviles incluyen malware, ataques de phishing, sacas y fuga de información. Por lo general, siempre se hace uso de técnicas de Ingeniería Social para engañar a los usuarios y consumar el ataque.

En este punto vamos a tratar las amenazas de malware sobre sistemas operativos móviles Android.

La plataforma móvil de Google se encuentra en el mercado desde septiembre del 2008 y a partir de ese momento ha tenido un crecimiento importante, llegando hoy en día a ser la plataforma móvil líder en teléfonos inteligentes. Como consecuencia, en la actualidad, es una de las plataformas para la cual más códigos maliciosos están apareciendo, explotando ciertas características presentes en la arquitectura del sistema y sus repositorios de aplicaciones.

Android dispone de su market de aplicaciones oficial Google Play al que se accede desde los dispositivos móviles usando la aplicación correspondiente. Google Play se encuentra protegido por diferentes mecanismos encargados de escalera las aplicaciones publicadas, en busca de signos que denoten un posible malware para proceder a su eliminación. A parte de este mecanismo de protección, los propios dispositivos Android disponen del servicio Package Verification, encargado de realizar algunas comprobaciones en las aplicaciones que van a ser instaladas. Incluso una vez instaladas es capaz de alertar al usuario de aquellas que son potencialmente dañinas, como pueden ser backdoors, phising, spyware, etc.

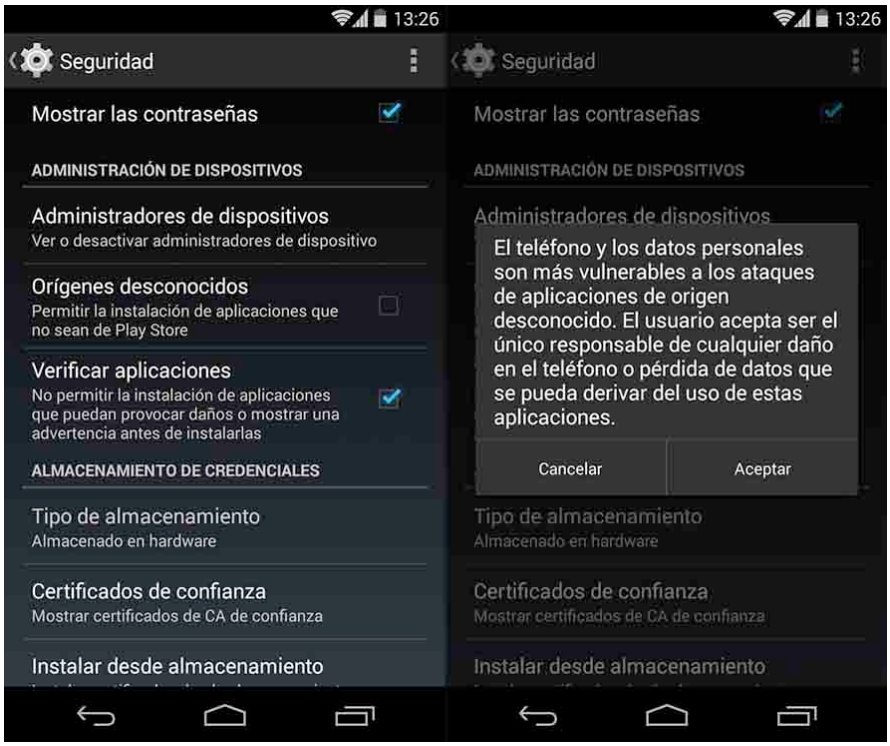
Google Play incorpora una serie de controles como por ejemplo no permitir que varias aplicaciones tengan la misma imagen de avatar o el mismo nombre. Así mismo, bloquea la publicación de aplicaciones en los siguientes casos:

- Aplicaciones que incluyen contenido ilegal.
- Aplicaciones que facilitan juegos de apuestas reales.
- Aplicaciones que incluyen contenido de promoción del odio.
- Aplicaciones que incluyen pornografía.
- Aplicaciones que incluyen violencia real gratuita.

Los markets de aplicaciones no se reducen al oficial, existe gran cantidad de mercados que podrían denominarse como "oficiales" para otras cosas, como puede ser el de Amazon AppStore o Samsung Apps, AppBrain, GetJar, AppsLib, Aptoide, etc. Algunos de ellos pueden estar preinstalados en algunos dispositivos.

La configuración por defecto de Android no permite la instalación desde estos orígenes y es necesario activar la opción "Orígenes desconocidos" en Ajustes del dispositivo. A partir de ese momento el usuario del terminal podrá instalar cualquier aplicación cuya procedencia sea distinta de Google Play.

No es recomendable tener activada esta opción, ya que si con los mecanismos de protección que dispone Google Play todavía se puede encontrar malware, desde fuentes no fiables el riesgo será mucho mayor.



Activación de orígenes desconocidos de apps en un dispositivo Android.

De hecho, el método de propagación de amenazas para Android suele ser a través de cuentas de desarrolladores falsas que publican aplicaciones maliciosas en el Android Market o a través de repositorios de aplicaciones no oficiales.

Para que nos hagamos una idea, si observamos algunos datos estadísticos de principios de año, los atacantes lograron subir más de 340 troyanos clicker de sitios pornográficos en la tienda Google Play en tan solo 7 meses (desde agosto de 2015 hasta febrero de 2016), y el número promedio de descargas alcanzó los 3.600 por cada app falsa.

Por tanto, podemos encontrarnos con diferentes tipos de malware como adware, phishing, spyware, RAT, keyloggers, Tap-jacking, clickers, ransomware, etc.

Uno de los acontecimientos más importantes del tercer trimestre de 2016 fue la aparición del juego Pokemon GO. Los delincuentes, no podían dejar pasar desapercibida la popularidad del nuevo juego y trataron de utilizarla para sus propios fines. Su técnica más frecuente era añadir código malicioso al juego original y propagar la aplicación maliciosa a través de tiendas de terceros. De esta manera, por ejemplo, se distribuía el troyano bancario Trojan-Banker.AndroidOS.Tordow, que utiliza una vulnerabilidad en el sistema para obtener permisos de root en el dispositivo. Una vez obtenidos los privilegios de root, este troyano se protege contra su eliminación y además puede robar las contraseñas almacenadas en el navegador.

Análisis de datos

Estructura del sistema de archivos

Android es un sistema operativo basado en Unix y, como tal, estructura sus archivos en secciones separadas, a las que asigna unos permisos de acceso y escritura, de tal forma que la accesibilidad de los archivos se determina por los permisos de cada uno de los archivos concretos, y del directorio o directorios que forman parte de su ruta de acceso, y de la sección en la que éstos se ubican.

Cada una de estas secciones, a las que denominaremos particiones, es absolutamente independiente del resto, lo que permite aislarlas entre sí y tratarlas como si fueran sistemas físicamente separados, lo que en realidad no tiene porqué ocurrir.

Cuando el sistema monta una determinada partición, éste le asigna unas determinadas opciones de montaje que incluyen tanto el modo de acceso (lectura y escritura, lectura exclusiva), como otras propiedades (sistema de ficheros que se implementa, posibilidad de ejecutar archivos de root, etc) y que no pueden modificarse posteriormente, lo que hace imposible, por ejemplo, escribir un archivo en una partición montada como de lectura exclusiva.

Aunque la organización de cada dispositivo depende finalmente del fabricante del mismo y de la implementación concreta que éste realice de Android, así com de la versión del sistema operativo, lo cierto es que todos comparten una estructura básica de particiones que incluye al menos las siguientes: system, data y sys.

La partición system contiene los programas y configuraciones que el fabricante u operador móvil suministra inicialmente con el teléfono, siendo su contenido montado por el sistema en modo de lectura exclusiva, lo que indica que el usuario puede leer los archivos que contiene pero no puede modificarlos, con independencia de los permisos asignados a los mismos.

El directorio app incluye las aplicaciones de sistema, tales como el lanzador de aplicaciones, las aplicaciones de contactos y de telefonía (en caso que el dispositivo soporte la realización y recepción de llamadas), el calendario, reproductor de música, etc.

El directorio etc contiene las configuraciones estáticas del sistema, así como los sonidos de notificaciones y tonos de llamada que se incluyen por defecto (que se ubican en el directorio /system/media/audio).

Los directorios bin y xbin contienen archivos binarios y scripts que utiliza el sistema para realizar algunas tareas, así como los scripts que permiten implementar algunos de los comandos del sistema, tales como el propio intérprete de comandos (habitualmente sh o bash).

En la partición de datos se encuentran los programas que instala el usuario y sus datos, así como los datos de las aplicaciones de sistema, se almacenan en la partición data, cuyo contenido es montado de tal forma que el sistema puede leer y escribir sus archivos, lo que permite la modificación dinámica de su contenido. El directorio app contiene las aplicaciones que instala el usuario, ya sea desde la propia Play Store de Google o desde alguna de las otras ubicaciones alternativas, y el directorio data almacena los archivos asociados a cada una de las aplicaciones.

La partición del kernel es montada habitualmente a partir de la carpeta sys, contiene el kernel del sistema, así como los módulos y librerías asociados a éste y los datos y archivos de cada uno de los dispositivos, tales como la propia CPU.

Análisis con Oxygen Forensic

En caso de querer analizar las aplicaciones de nuestro dispositivo para detectar el malware o simplemente realizar un análisis de los datos de forma rápida y cómoda, la herramienta seleccionada es Oxygen Forensic. Y el procedimiento es igual que con los dispositivos iOS, como comentamos en el Análisis de datos en iOS.

La única diferencia en el procedimiento respecto a iOS, es que en el caso de los terminales Android, la herramienta Oxygen Forensic Suite realiza el proceso de Rooteo del dispositivo al inicio, de forma automática.

INFORME FINAL Y CONCLUSIONES

Dispositivos móviles con Android OS

Para concluir con este trabajo de como debería realizarse un análisis de seguridad sobre dispositivos con Android OS, debemos tener en cuenta diferentes aspectos. En el caso de los smartphones con Android OS, es complicado definir detalladamente los procedimientos a realizar debido al problema de fragmentación de Android. A principios de 2016 un estudio de OpenSignal revelaba que existen más de 24.000 modelos diferentes de dispositivos entre smartphones y tabletas. Pero de forma genérica podemos realizar una aproximación de lo que sería esta metodología.

En primer lugar, hay que tener en cuenta que las principales amenazas, o al menos las que más abundan, están relacionadas directamente con algún tipo de malware, que mediante diferentes técnicas sobretodo de ingeniería social, consiguen instalarse en los dispositivos móviles para vulnerar la seguridad de estos. Por tanto para detectar este tipo de aplicaciones y protegerse de ellas, a parte de seguir algunas recomendaciones de sentido común, es necesario utilizar herramientas para este objetivo, como pueden ser antivirus y antimalware.

Respecto al tema de las vulnerabilidades públicas de Android OS, podemos hacer una búsqueda de la versión instalada en el dispositivo a analizar en la web de CVE Details https://www.cvedetails.com y obtendremos un listado de las que afectan a esta versión, su gravedad e impacto. Estos datos los podemos contrastar con la web oficial de Android, donde aparecen todas las vulnerabilidades reportadas. Existen aplicaciones que realizan este procedimiento de forma automática, como la app Android VTS, que instalamos y ejecutamos en el dispositivo móvil y nos indica que vulnerabilidades afectan a este y la gravedad de estas.

Con la versión de Android también podemos buscar que herramientas para hacer Root al dispositivo y de qué manera se puede lograr.

De la misma forma, mediante diferentes aplicaciones se puede saber si el smartphone en análisis está Rooteado. Ya que existen diferentes métodos de rooteo de forma remota que han afectado a gran cantidad de dispositivos y los usuarios propietarios de los mismos, no son conscientes de ello. Conseguir tener un terminal Rooteado sería útil para poder tomar el control de un dispositivo y para realizar un análisis de sus datos más exhaustivamente.

Realizar una detección de los servicios que están en escucha en el dispositivo a través de la red, es también un procedimiento que nos ayudará a detectar aplicaciones que podrían ser un agujero de seguridad. Habrá que estudiar cada caso de forma detallada, según el tipo de aplicación, su función y su configuración.

Analizar la configuración del dispositivo, es decir las diferentes opciones de Android OS en materia de configuración. Como la conectividad, la instalación de apps de fuentes no seguras, activación de sistemas de seguridad de acceso, desactivar el modo depuración USB, etc. Todo esto debemos hacerlo de forma manual, ya que una configuración incorrecta podría facilitar a un atacante acceder al dispositivo, ya sea a través de una app infectada con malware desde un origen no seguro o mediante un ataque de red.

Para finalizar, habrá que realizar un análisis de las aplicaciones que hay instaladas para detectar posible malware. Mediante herramientas como Oxygen Forensic podremos realizar un análisis de las apps para verificar que estas no están infectadas o no son dañinas.

Dispositivos móviles con iOS

En este caso, vamos a resumir la metodología para smartphone de la compañía Apple Inc. que solamente pueden tener iOS como sistema operativo.

Podemos afirmar que la principal amenaza externa para los smartphones con iOS viene cuando el usuario realiza el jailbreak en su dispositivo.

Otra de las amenazas importantes, al igual que ocurre con Android, es el malware. Es cierto que las medidas de seguridad y control de las que dispone la App Store son eficaces y ofrecen una tranquilidad al usuario respecto a otras plataformas, pero aún así algunas apps se han podido saltar estos controles e infectar a los dispositivos.

Por otro lado existen otro tipo de amenazas que tienen como objetivo los terminales iOS, pero que utilizan los equipos donde estos están “pareados” para propagarse e infectarlos mediante cable, cuando estos se conectan al equipo en cuestión. También de esta forma, atacando a los equipos “pareados” se han conseguido sustraer datos de la copia de seguridad almacenada en los equipos o en iCloud.

Para comenzar con la metodología debemos obtener la versión y modelo del dispositivo. Con estos datos en la página de CVE Details realizar la búsqueda correspondiente, que podemos contrastar con la web de actualizaciones de seguridad de Apple, y obtendremos un listado con las vulnerabilidades que afectan al dispositivo a analizar. Con estos datos podemos saber a que riesgos está expuesto el smartphone, con la versión de iOS que tiene instalada.

Con estos datos podremos saber si existen herramientas para realizar el Jailbreak al dispositivo, y sobretodo si existe la opción del jailbreak remoto.

Comprobaremos que el dispositivo no tiene hecho el Jailbreak, ya que podría tener problemas de seguridad relacionados con este estado, por eso no se recomienda.

Es importante realizar un análisis de los servicios en ejecución que están escuchando en el dispositivo a través de la red, ya que estaríamos abriendo puertos de entrada desde el exterior, es también un procedimiento que nos ayudará a detectar aplicaciones que podrían ser un agujero de seguridad. Habrá que estudiar cada caso de forma detallada, según el tipo de aplicación, su función y su configuración.

De forma manual, debemos comprobar la configuración de las diferentes opciones del sistema operativo en cuanto a conectividad, control de acceso, cifrado, configuración del navegador, etc, ya que una configuración incorrecta podría facilitar a un atacante acceder al terminal.

Y por último, mediante herramientas de análisis de aplicaciones móviles, en este caso recomendamos Oxigen Forensic, realizar un análisis de las aplicaciones instaladas en el dispositivo para detectar algún tipo de malware.

Conclusiones

En el análisis de vulnerabilidades y la auditoria de seguridad en Smartphones no es sencillo establecer una metodología estricta a seguir. La realidad es que existe un gran número de modelos y marcas de smartphones, aunque la gran mayoría utilicen los sistemas operativos Android e iOS, con una gran cantidad de versiones y variantes, que a han ido surgiendo a lo largo de estos años.

Esto tiene el inconveniente de que cada Investigador utilice técnicas y habilidades que se puede adquirir en base a la experiencia. Sin embargo, en este trabajo se pretende dar algunas pautas que pueden servir de guía para el analista que desea realizar un análisis sobre smartphones, concretamente sobre terminales con sistema operativo iOS y Android OS.

Trabajo futuro

Para completar este proyecto sería recomendable la realización de varias pruebas de concepto en un laboratorio, con al menos dos dispositivos diferentes con los sistemas operativos iOS y Android OS respectivamente. Estas pruebas ayudarían a demostrar los procedimientos desarrollados en este trabajo y a dar a conocer el concepto práctico de las técnicas estudiadas.

BIBLIOGRAFÍA

Uso general

"Hacking de dispositivos iOS: iPhone & iPad", 2ª edición, Chema Alonso, 0xWORD Computing SL

"Malware en Android": Discovering, Reversing & Forensic", Miguel Ángel Garcia del Moral, 0xWORD Computing SL

https://www.oxygen-forensic.com/es/

http://www.welivesecurity.com/wp-content/uploads/2014/01/malware_en_dispositivos_moviles.pdf

http://www.welivesecurity.com/wp-content/uploads/2015/11/Guia_respuesta_infeccion_malware_ESET.pdf

http://www.welivesecurity.com/la-es/2016/02/24/troyanos-clicker-google-play/

https://www.certsi.es/sites/default/files/contenidos/estudios/doc/situacion_del_malware_para_android.pdf

http://faqsandroid.com/estructura-del-sistema-de-ficheros-en-android/

http://blog.uptodown.com/wp-content/uploads/android-origenes-desconocidos.jpg

https://buguroo.com/es/tecnicas-de-evasion-de-bloqueo-y-rooteo-en-dispositivos-android

https://www.certsi.es/blog/herramientas-forense-moviles

http://www.thingsupsecurity.com/2013/03/analisis-forense-en-android-parte-i.html#more

http://www.sahw.com/wp/archivos/2010/05/23/analisis-forense-de-telefonos-basados-en-sistemas-android/

http://www.hackcave.net/2015/10/tutorial-hackingbypassing-android.html

http://www.sniferl4bs.com/2014/11/analisis-forense-android-ii-rompiendo.html

http://www.androidsis.com/como-rootear-tu-android-con-root-master-sin-necesidad-de-pc/

https://www.incibe.es/protege-tu-empresa/que-te-interesa/proteccion-movilidad-conexiones-inalambricas

https://es.m.wikipedia.org/wiki/Seguridad_en_telefon%C3%ADa_móvil

http://www.welivesecurity.com/la-es/2015/02/25/que-es-un-0-day/

http://www.kaspersky.es/internet-security-center/internet-safety/smartphones

http://www.20minutos.es/noticia/830501/0/virus/telefonos/moviles/

http://www.ticbeat.com/seguridad/5-amenazas-actuales-peligrosas-dispositivos-moviles/

http://support.kaspersky.com/sp/viruses/general/614

http://www.pandasecurity.com/spain/mediacenter/consejos/10-consejos-para-evitar-ataques-de-phishing/

https://ofiseg.wordpress.com/2012/05/24/vias-de-infeccion-para-dispositivos-moviles-prevencion-y-mitigacion/

http://andro4all.com/2015/08/android-fragmentacion-2015

https://www.sophos.com/es-es/medialibrary/PDFs/other/sophos-mobile-security-threat-report.pdf?la=es-ES

http://www.cromo.com.uy/cuales-seran-las-principales-amenazas-2016-n711688

http://www.mcafee.com/es/resources/reports/rp-threats-predictions-2016.pdf

http://www.pandasecurity.com/spain/mediacenter/src/uploads/2016/05/Pandalabs-2016-T1-LR-ES.pdf

http://www.androidpit.es/nfc-que-es-para-que-sirve

http://muyseguridad.net/2011/11/11/informe-seguridad-tecnologia-nfc/

Definición del SO iOS

https://es.wikipedia.org/wiki/IOS

Metodología para el análisis de vulnerabilidades

http://searchdatacenter.techtarget.com/es/consejo/Como-realizar-una-auditoria-de-seguridad-de-siguiente-generacion-para-redes

http://searchdatacenter.techtarget.com/es/consejo/Como-realizar-una-auditoria-de-seguridad-de-siguiente-generacion-para-redes

http://noticiasseguridad.com/tecnologia/como-hacer-analisis-de-vulnerabilidades-informaticas/

http://www.antpji.com/antpji2013/index.php/articulos2/111-auditoria-de-seguridad-informatica

Arquitectura de seguridad de iOS

https://hacking-etico.com/2014/07/01/ios-hacking-arquitectura-de-seguridad-en-ios/

http://eve-ingsistemas-u.blogspot.com.es/2012/04/sistemas-operativos-moviles-ios.html

https://www.apple.com/es/business/docs/iOS_Security_Guide.pdf

Definición del SO Android

http://www.xatakandroid.com/sistema-operativo/que-es-android

https://es.wikipedia.org/wiki/Android

Arquitectura del SO Android

http://www.androidcurso.com/index.php/99

https://columna80.wordpress.com/2011/02/17/arquitectura-de-android/

Sistema de archivos del SO Android

http://andro4all.com/2014/04/f2fs-vs-ext4

http://androidos.readthedocs.io/en/latest/data/detalles_tecnicos/

Arquitectura de seguridad del SO Android

https://media.readthedocs.org/pdf/androidos/latest/androidos.pdf

http://source.android.com/tech/security/

http://e-archivo.uc3m.es/bitstream/handle/10016/18102/PFC_Cristian_Madero_Garcia.pdf?sequence=1

http://paradigma.uniandes.edu.co/images/sampledata/PARADIGMA/ediciones/Edicion7/Numero1/Articulo1/mendez-sanchez_ed7-1.pdf

Escaneo de dispositivos móviles iOS

https://netting.wordpress.com/2013/08/20/identificacion-de-dispositivos-ios/

https://www.reddit.com/r/jailbreak/wiki/escapeplan/guides/jailbreakcharts

https://support.apple.com/es-es/HT201296

https://www.theiphonewiki.com/

Evaluación y acceso al sistema en iOS

https://en.wikipedia.org/wiki/XcodeGhost

https://developer.apple.com/library/content/documentation/IDEs/Conceptual/AppStoreDistributionTutorial/CreatingYourTeamProvisioningProfile/CreatingYourTeamProvisioningProfile.html

http://myicloud.info/brute-forcing-ios-9-2-1-passcode/

http://www.elladodelmal.com/2013/06/saltar-el-passcode-de-un-terminal-ios.html

http://static.usenix.org/events/woot10/tech/full_papers/Aviv.pdf

http://www.idownloadblog.com/2016/01/19/hide-notification-preview-text-email-notification-lock-screen/

http://beebom.com/ios-10-vs-ios-9-comparison/

https://www.ccn-cert.cni.es/series-ccn-stic/guias-de-acceso-publico-ccn-stic/13-ccn-stic-455-seguridad-en-iphone/file.html

Man in the Middle: Fair Play MitM

http://www.seguridadapple.com/2016/03/acedeceiver-un-nuevo-tipo-de-malware-en.html

Otros ataques

http://researchcenter.paloaltonetworks.com/2016/03/acedeceiver-first-ios-trojan-exploiting-apple-drm-design-flaws-to-infect-any-ios-device/

http://ioshacker.com/apps/system-security-info-app-detects-malware-lists-useful-information-ios-devices

http://www.seguridadapple.com/2015/01/ese-troyano-puede-infectar-tu-iphone.html

Escaneo de dispositivos móviles Android

http://foromoviles.com/cuales-diferentes-tipos-de-bloqueo-en-android/

http://www.groovypost.com/howto/play-android-6-marshmallow-hidden-flappy-bird-game/

https://buguroo.com/es/tecnicas-de-evasion-de-bloqueo-y-rooteo-en-dispositivos-android

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

https://es.kingoapp.com/root-tutorial/

http://hexamob.com/es/como-rootear-movil-o-tableta-android/metodo-de-rooteo-srsroot/

http://www.androidsis.com/como-rootear-tu-terminal-android-con-framaroot/

https://source.android.com/security/bulletin/

http://blog.checkpoint.com/2016/08/07/quadrooter/

Evaluación y accesos al sistema Android

https://es.wikipedia.org/wiki/Android_rooting

https://www.certsi.es/blog/introduccion-analisis-forense-en-moviles

https://opensignal.com/reports/2015/08/android-fragmentation/

https://www.certsi.es/blog/herramientas-forense-moviles

http://www.elandroidelibre.com/2015/01/como-activar-el-modo-depuracion-usb-en-android.html

http://www.androidpit.es/que-es-la-depuracion-usb

http://www.androidpit.es/que-es-adb-comandos-mas-importantes

ANEXO

Medidas de prevención y recomendaciones

Utilizamos nuestros teléfonos para una variedad mucho muy amplia de actividades. Por este motivo, debemos adoptar precauciones sensatas para garantizar que nuestros teléfonos e información están a salvo de ataques de malware y cibercriminales.

- Para el uso de dispositivos móviles se deberán seguir las siguientes recomendaciones y medidas de prevención de forma genérica:
- Al reciclar o vender un teléfono móvil, eliminar todo el contenido personal de las memorias.
- Comprar e instalar aplicaciones y Software en páginas oficiales y/o de confianza.
- Mantener actualizado el sistema operativo y las aplicaciones.
- Apagar el móvil por la noche cuando no está siendo utilizado
- Usar contraseñas alfanuméricas o PIN para el acceso y tras la inactividad de los dispositivos.
- Evitar proporcionar información financiera y personal vía correo electrónico, conversaciones telefónicas o por SMS.
- Activar la encriptación de datos y cifrado del dispositivo y sus memorias externas.
- Hacer copias de seguridad de los datos y del sistema para restablecer el sistema en caso de fallos o pérdidas de información.
- Usar servicios de localización online, para permitir el borrado de datos en caso de robo o extravío.
- Conectarse a redes Wi-Fi conocidas y seguras.
- Mantener apagado el Bluetooth y sólo mantenerlo activo durante su utilización.
- Instalar un software antivirus y/o antimalware.
- Precaución con la descarga y recepción de archivos.
- Precaución al navegar por Internet, y sobre todo con los enlaces de correos, mensajes SMS o en redes sociales.
- No eliminar las protecciones del sistema.
- Cumplir con la directiva de contraseñas y usar un gestor para las mismas.
- Acceder a la red empresarial a través de redes privadas o VPN.
- Instalar aplicación de seguridad y rastreo que permita la localización y borrado del dispositivo.

- No liberar dispositivos mediante técnicas de root o jailbreak.
- Supervisa el comportamiento y permisos de las aplicaciones en tu smartphone.
- Concienciación del usuario para evitar comportamientos que puedan provocar riesgos.

Plan de respuesta y mitigación en caso de infección por malware

Debido a la naturaleza de las amenazas, existe la probabilidad de padecer las consecuencias de un incidente, incluso a veces contando con controles de seguridad implementados. Estos planes se consideran actividades preventivas y reactivas, de manera que puedan evitarse las infecciones por malware o, en su defecto, que de presentarse sus consecuencias sean las mínimas aceptables. Cuando se trata de incidentes relacionados con malware, se pueden seguir actividades que contribuyen a una efectiva respuesta y rápida recuperación. A continuación describiremos brevemente las fases de un Plan de Respuesta a Incidentes de Seguridad, para atender un incidente de infección por malware.

1) Identificar la infección: Un incidente de seguridad relacionado con malware puede ser detectado de diferentes maneras y con distintos niveles de detalle y para conseguirlo necesitaremos herramientas para su detección.

2) Determinar su alcance: Luego de la identificación de una infección por malware, es necesario determinar que parte del sistema que ha sido comprometido y de qué manera, con el propósito de conocer el alcance de la infección y el impacto que puede representar. A partir del tipo de malware y su comportamiento, también es posible determinar saber si se ha filtrado información sensible, si se han visto comprometidos datos corporativos o privados. En función del malware que nos haya afectado, tendríamos estos posibles efectos y medidas a adoptar:

i. Robo de datos de contactos. Es necesario avisar a nuestros contactos e indicarles que es posible que reciban alguna comunicación sospechosa, probablemente en nuestro nombre.

ii. Robo de contraseñas. Será necesario el cambio de nuestra contraseña de los servicios que se haya usado desde este dispositivo, ya que podrían haber sido robadas. Con el robo de credenciales se podrá suplantar la identidad en el servicio en cuestión.

iii. Robo de datos bancarios. La única forma de mitigar esta acción es informar a nuestro banco de lo sucedido, modificar las credenciales bancarias, y en caso de que hayan usado ya esas credenciales con fines fraudulentos, realizar una denuncia ante las Fuerzas y Cuerpos de Seguridad del Estado.

iv. Robo de información empresarial. En el caso de que el dispositivo móvil se utilice con fines profesionales, el malware podría robar información relativa a la empresa. La forma de mitigar esta acción es comunicárselo a la empresa para que tome las medidas necesarias.

3) Mantener la continuidad del negocio: En el ámbito corporativo, luego de conocer el alcance de la infección, se podrá determinar si información sensible o equipos críticos se han visto afectados. En función de este resultado, se podrán tomar decisiones para continuar correctamente con las operaciones de la compañía.

4) Contener las acciones maliciosas: Las estrategias de contención pueden variar en función del incidente y de los lineamientos establecidos por los equipos de respuesta, lo que a su vez depende del tipo de malware que afecte a la organización. Por otro lado, la identificación del vector de ataque resulta fundamental para contener los estragos generados por un código malicioso y evitar su propagación. Los vectores más comunes son a través de medios externos(tarjetas SD), explotación de vulnerabilidades en el software y sitios web, archivos adjuntos a correos electrónicos y enlaces a sitios que alojan malware.

5) Erradicar la infección y el vector de ataque: A partir de la identificación del método de propagación, será necesario desinfectar el dispositivo mediante un antivirus o antimalware. También es obligatorio llevar a cabo algunas acciones que mitiguen de manera específica el vector de ataque, por ejemplo el filtrado de correo electrónico y análisis de los mensajes y adjuntos y la modificación de los sistemas operativos para evitar la ejecución de programas sospechosos.

6) Recuperar el control: La fase de recuperación se presenta luego de que un incidente por malware ha sido contenido y de que se han identificado y mitigado las vulnerabilidades que fueron explotadas. Llegado este punto, se confirma que los sistemas se encuentran funcionando de manera normal y que el malware ha sido removido para evitar incidentes similares. La recuperación puede incluir acciones como la restauración de sistemas operativos y respaldos, el reemplazo de archivos infectados, la instalación de parches de seguridad y actualizaciones, el cambio de contraseñas en los sistemas, etc.