

CASO PRÁCTICO: FUNDAMENTOS DE SEGURIDAD CLOUD E INFRAESTRUCTURAS INDUSTRIALES

Alumno: Gengis Rovi

1. INTRODUCCIÓN

El presente informe detalla el procedimiento técnico y metodológico llevado a cabo para resolver dos escenarios prácticos de seguridad y despliegue de infraestructuras.

La primera sección aborda un análisis forense digital sobre un dispositivo móvil Android (evidencia `data.dd`), con el objetivo de esclarecer las actividades sospechosas del sujeto "Mr. X". Se ha utilizado una metodología de extracción física simulada y análisis de sistemas de ficheros y bases de datos SQLite en un entorno macOS.

La segunda sección detalla la implementación de arquitecturas basadas en contenedores Docker, aplicando buenas prácticas de seguridad como el uso de usuarios no privilegiados, persistencia de datos mediante volúmenes y orquestación de servicios mediante Docker Compose.

2. ANÁLISIS FORENSE DE DISPOSITIVOS MÓVILES (CASO MR. X)

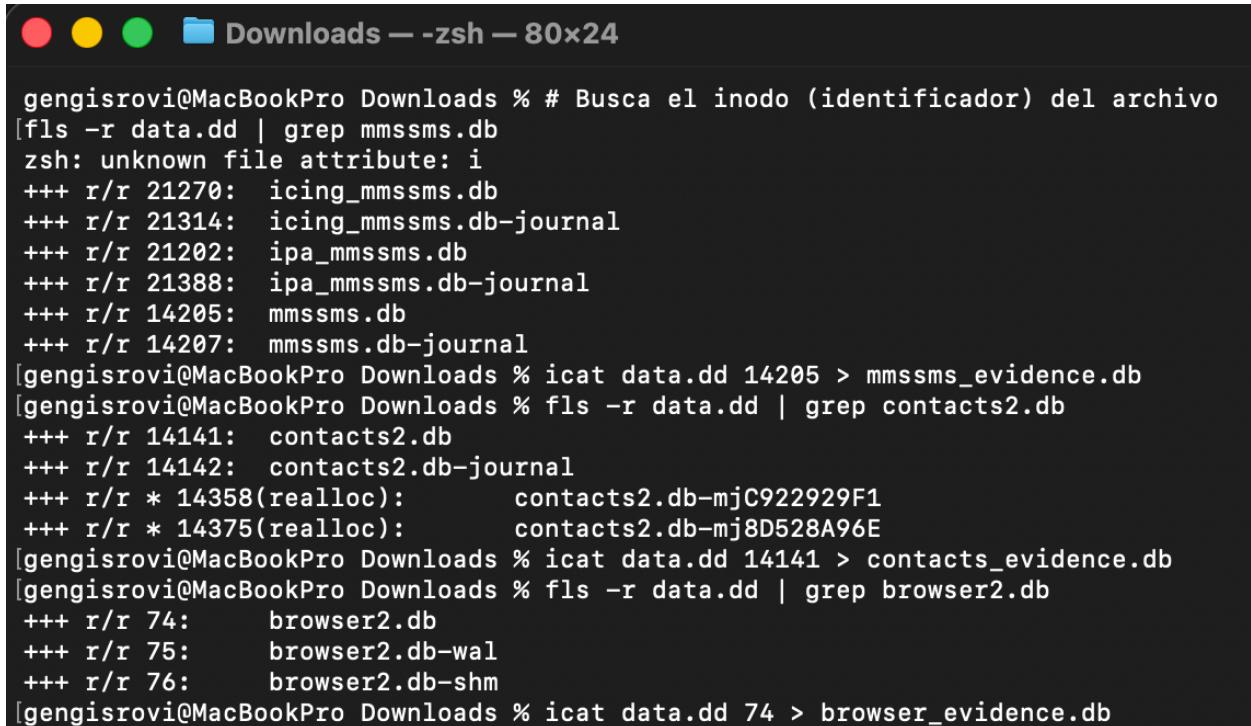
2.1. Metodología y Herramientas

Para el análisis del archivo de evidencia `data.dd`, se ha procedido a trabajar sobre una estación de trabajo macOS. Dado que el sistema de ficheros de Android (típicamente EXT4) no es nativo en este entorno, se han utilizado herramientas de análisis forense para la localización de inodos y extracción de ficheros sin alterar la integridad de la evidencia.

- **Evidencia:** Archivo de imagen `data.dd`.
- **Herramientas:**
 - **The Sleuth Kit (TSK):** Herramientas `mmls` (para particionado), `fls` (para listar archivos) y `icat` (para extracción de contenido).
 - **DB Browser for SQLite:** Para el análisis de las bases de datos extraídas (`.db`).
 - **Terminal macOS:** Para la ejecución de comandos y filtrado con `grep`.

2.2. Preparación y Análisis de Particiones

En primer lugar, se analizó la estructura de particiones de la imagen para localizar la partición de datos de usuario (`/data`), donde reside la información crítica de Android como SMS y contactos.



```
gengisrovi@MacBookPro Downloads % # Busca el inodo (identificador) del archivo
[fls -r data.dd | grep mmssms.db
zsh: unknown file attribute: i
+++ r/r 21270: icing_mmssms.db
+++ r/r 21314: icing_mmssms.db-journal
+++ r/r 21202: ipa_mmssms.db
+++ r/r 21388: ipa_mmssms.db-journal
+++ r/r 14205: mmssms.db
+++ r/r 14207: mmssms.db-journal
[gengisrovi@MacBookPro Downloads % icat data.dd 14205 > mmssms_evidence.db
[gengisrovi@MacBookPro Downloads % fls -r data.dd | grep contacts2.db
+++ r/r 14141: contacts2.db
+++ r/r 14142: contacts2.db-journal
+++ r/r * 14358(realloc): contacts2.db-mjC922929F1
+++ r/r * 14375(realloc): contacts2.db-mj8D528A96E
[gengisrovi@MacBookPro Downloads % icat data.dd 14141 > contacts_evidence.db
[gengisrovi@MacBookPro Downloads % fls -r data.dd | grep browser2.db
+++ r/r 74: browser2.db
+++ r/r 75: browser2.db-wal
+++ r/r 76: browser2.db-shm
[gengisrovi@MacBookPro Downloads % icat data.dd 74 > browser_evidence.db
```

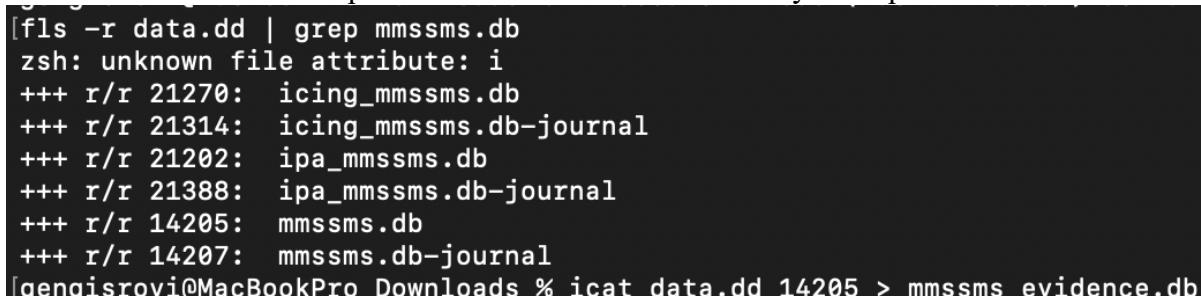
Figura 1. Análisis de la tabla de particiones de la imagen forense.

2.3. Resolución de las Incógnitas del Caso

2.3.1. Trama, Cómplices y Dinero (Análisis de SMS)

Para determinar las intenciones de Mr. X, se procedió a localizar y extraer la base de datos de mensajería `mmssms.db`, ubicada teóricamente en `/data/data/com.android.providers.telephony/databases/`.

Se utilizó el comando `fls` para localizar el inodo del archivo y `icat` para su extracción:



```
[fls -r data.dd | grep mmssms.db
zsh: unknown file attribute: i
+++ r/r 21270: icing_mmssms.db
+++ r/r 21314: icing_mmssms.db-journal
+++ r/r 21202: ipa_mmssms.db
+++ r/r 21388: ipa_mmssms.db-journal
+++ r/r 14205: mmssms.db
+++ r/r 14207: mmssms.db-journal
[gengisrovi@MacBookPro Downloads % icat data.dd 14205 > mmssms_evidence.db
```

Figura 2. Localización y extracción forense de la base de datos de mensajería.

Tras abrir el archivo `mmssms_evidence.db` en DB Browser y analizar la tabla `sms`, se obtuvieron los siguientes hallazgos:

	thread_id	address	person	date	date_sent	protocol	read	status	type	reply_path_present	subject	body	service_center	locked	sub_id	error_code	creator	s
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	650551212	NULL	185731916584	1857279517000	0	1	-1	1	0	NULL	Yumm! Pie & la Android mode!	NULL	0	-1	0	com.android.mms	
2	2	650-555-1111	NULL	1857387006800	0	NULL	1	-1	2	NULL	NULL	hey Matt I need help cracking some passwords	NULL	0	1	0	com.android.mms	
3	2	650551111	1	1857387033031	18573246382000	0	1	-1	1	0	NULL	Not sure if I will be able to help. What do you need?	NULL	0	-1	0	com.android.mms	
4	2	650-555-1111	NULL	1857387054594	0	NULL	1	-1	2	NULL	NULL	I got a shadow file from one of the 3 letters agency	NULL	0	1	0	com.android.mms	
5	2	650551111	1	1857387056141	1857324667000	0	1	-1	1	0	NULL	dude, you are playing with fire	NULL	0	-1	0	com.android.mms	
6	2	650551111	1	1857387077283	1857324676000	0	1	-1	1	0	NULL	I don't think I can help, but I know somebody who may	NULL	0	-1	0	com.android.mms	
7	2	650-555-1111	NULL	1857387088409	0	NULL	1	-1	2	NULL	NULL	who is it?	NULL	0	1	0	com.android.mms	
8	2	650551111	1	1857387108804	1857324704000	0	1	-1	1	0	NULL	Danny Rand, here is his number 650-555-2222	NULL	0	-1	0	com.android.mms	
9	2	650-555-1111	NULL	1857387116477	0	NULL	1	-1	2	NULL	NULL	thanks bro	NULL	0	1	0	com.android.mms	
10	3	650-555-2222	NULL	1857387144158	0	NULL	1	-1	2	NULL	NULL	hi Danny, this is Mr X, I need your help	NULL	0	1	0	com.android.mms	
11	3	650552222	2	1857387162689	1857324768000	0	1	-1	1	0	NULL	how did you get my number?	NULL	0	-1	0	com.android.mms	
12	3	650-555-2222	NULL	1857387188676	0	NULL	1	-1	2	NULL	NULL	Matt gave it to me	NULL	0	1	0	com.android.mms	
13	3	650552222	2	1857387202363	1857324801000	0	1	-1	1	0	NULL	What do you need?	NULL	0	-1	0	com.android.mms	
14	3	650-555-2222	NULL	1857387210415	0	NULL	1	-1	2	NULL	NULL	I need help cracking some hashes	NULL	0	1	0	com.android.mms	
15	3	650552222	2	1857387208449	1857324819000	0	1	-1	1	0	NULL	what kind of hashes?	NULL	0	-1	0	com.android.mms	
16	3	650-555-2222	NULL	1857387232810	0	NULL	1	-1	2	NULL	NULL	I don't know, I'm totally illiterate, hence that's why I need...	NULL	0	1	0	com.android.mms	
17	3	650-555-2222	NULL	1857387248816	0	NULL	1	-1	2	NULL	NULL	Illiterate I meant	NULL	0	1	0	com.android.mms	
18	2	650551111	1	1857387282281	1857324881000	0	1	-1	1	0	NULL	hey, did you contact Danny?	NULL	0	-1	0	com.android.mms	
19	2	650-555-1111	NULL	1857387318739	0	NULL	1	-1	2	NULL	NULL	Yeah, I'm talking with him right now	NULL	0	1	0	com.android.mms	
20	2	650551111	1	1857387328004	1857324927000	0	1	-1	1	0	NULL	cool, just checking	NULL	0	-1	0	com.android.mms	
21	3	650552222	2	1857387328408	1857324981000	0	1	-1	1	0	NULL	where did you get that data from?	NULL	0	-1	0	com.android.mms	
22	3	650-555-2222	NULL	18573873774219	0	NULL	1	-1	2	NULL	NULL	Hold on, brb	NULL	0	1	0	com.android.mms	
23	3	650-555-2222	NULL	1857365488385	0	NULL	1	-1	2	NULL	NULL	hey, I'm back	NULL	0	1	0	com.android.mms	
24	3	650-555-2222	NULL	1857365488787	0	NULL	1	-1	2	NULL	NULL	From a 3 letters agency, I prefer not disclosing	NULL	0	1	0	com.android.mms	
25	3	650552222	2	1857365488868	1857333087000	0	1	-1	1	0	NULL	you crazy? see what happened to Assange for trying to hel...	NULL	0	-1	0	com.android.mms	
26	3	650-555-2222	NULL	1857365488349	0	NULL	1	-1	2	NULL	NULL	That's a different story, that guy pissed off some big people...	NULL	0	1	0	com.android.mms	
27	3	650-555-2222	NULL	1857365536442	0	NULL	1	-1	2	NULL	NULL	can you help me or not?	NULL	0	1	0	com.android.mms	
28	3	650552222	2	1857365878690	1857333174000	0	1	-1	1	0	NULL	not sure yet. I need to see the hashes	NULL	0	-1	0	com.android.mms	
29	3	650-555-2222	2	18573658610700	1857333090000	0	1	-1	1	0	NULL	if it is double, I can let you rent my GPU farm to run the ...	NULL	0	-1	0	com.android.mms	
30	3	650-555-2222	NULL	1857365862107	0	NULL	1	-1	2	NULL	NULL	How much?	NULL	0	1	0	com.android.mms	
31	3	650552222	2	1857365864438	1857333083000	0	1	-1	1	0	NULL	\$10k a week	NULL	0	-1	0	com.android.mms	

Figura 3. Contenido de los mensajes SMS incriminatorios.

- **¿Qué tramaba Mr. X?:** Descifrar algunas contrasenas de una compania de 3 letras
- **¿A quién le pidió ayuda inicialmente?:** Le pide ayuda a Matt, su numero seria 650-555-1111, para descifrar algunas contrasenas.
- **¿Quién le intentó ayudar y por cuánto dinero?:** Danny Rand y su numero seria 650-555-2222, intento ayudar por 10k a la semana

2.3.2. Enlace del Fichero y Navegación

Para localizar el enlace compartido y analizar la navegación, se extrajo la base de datos del historial del navegador, típicamente `browser2.db` o `History` (Chrome), ubicada en `/data/data/com.android.browser/databases/`.

The screenshot shows the DB Browser for SQLite interface. The main window displays the 'history' table from a database named 'browser_evidence.db'. The table has three columns: '_id', 'title', and 'url'. The data shows 13 rows of browser history, including visits to Google, Cyberhades, and various password cracking resources. The right panel shows the current row being edited, with mode set to 'Text'.

	_id	title	url
1	1	Google	https://www.google.com/webhp?source=android&hl=es
2	2	https://cyberhades.ams3.digitaloceanspaces.com/shadow	https://cyberhades.ams3.digitaloceanspaces.co...
3	3	Cyberhades	http://www.cyberhades.com/
4	4	Cyberhades	http://www.cyberhades.com/en/
5	5	Moving the blog images from Flickr to Digital Ocean Spaces	https://www.cyberhades.com/en/2019/02/03/m...
6	6	Hackaday Fresh hacks every day	https://hackaday.com/
7	7	Google	https://www.google.com/?gws_rd=ssl
8	8	http://www.google.com/	http://www.google.com/
9	9	https://www.google.com/?gws_rd=ssl#sbfbu=1&pl...	https://www.google.com/?gws_rd=ssl#sbfbu=1&...
10	10	how to crack a password - Google Search	https://www.google.com/search?...
11	11	How to Crack a Password	https://www.guru99.com/how-to-crack-passwo...
12	12	10 Most Popular Password Cracking Tools [Updated for ...	https://resources.infosecinstitute.com/10-popul...
13	13	https://resources.infosecinstitute.com/10-popular...	https://resources.infosecinstitute.com/10-popul...

Figura 4. Historial de navegación y localización del enlace.

- **¿Quién le pasó el enlace?:** Jessica con numero 6505554444, le compartio el enlace.
- **¿Cuál es el enlace?:** <https://cyberhades.ams3.digitaloceanspaces.com/shadow>
- **Historial y Cookies:** Se ha verificado la tabla de historial mostrando la actividad reciente del usuario, efectivamente se puede ver como Mr. x cuenta con el enlace en cuestion en el historial.

2.3.3. Identificación de Contactos y Llamadas

Para corroborar la identidad de los interlocutores, se analizó la base de datos `contacts2.db`, ubicada en `/data/data/com.android.providers.contacts/databases/`.

The screenshot shows the DB Browser for SQLite interface. The main window displays the 'view_contacts' table from a database named 'contacts2.db'. The table has ten columns: '_id', 'custom_ringtone', 'display_name_source', 'display_name', 'display_name_alt', 'phonetic_name', 'phonetic_name_style', 'sort_key', 'phonebook_label', and 'phonebook'. The data shows four rows of contacts, including Jessica Jones, Mat Murdock, Frank Castle, and Danny Rand. The right panel shows the current row being edited, with mode set to 'Text'.

	_id	custom_ringtone	display_name_source	display_name	display_name_alt	phonetic_name	phonetic_name_style	sort_key	phonebook_label	phonebook
1	1	NULL	40	Jessica Jones	Jones, Jessica	NULL	0	J	Jessica Jones	J
2	2	NULL	40	Mat Murdock	Murdock, Mat	NULL	0	M	Mat Murdock	M
3	3	NULL	40	Frank Castle	Castle, Frank	NULL	0	F	Frank Castle	F
4	4	NULL	40	Danny Rand	Rand, Danny	NULL	0	D	Danny Rand	D

DB Browser mostrando la tabla `calls` Figura 5. Registro de llamadas y lista de contactos.

Se cruzaron los números de teléfono de los SMS con esta base de datos para identificar los nombres reales de los implicados.

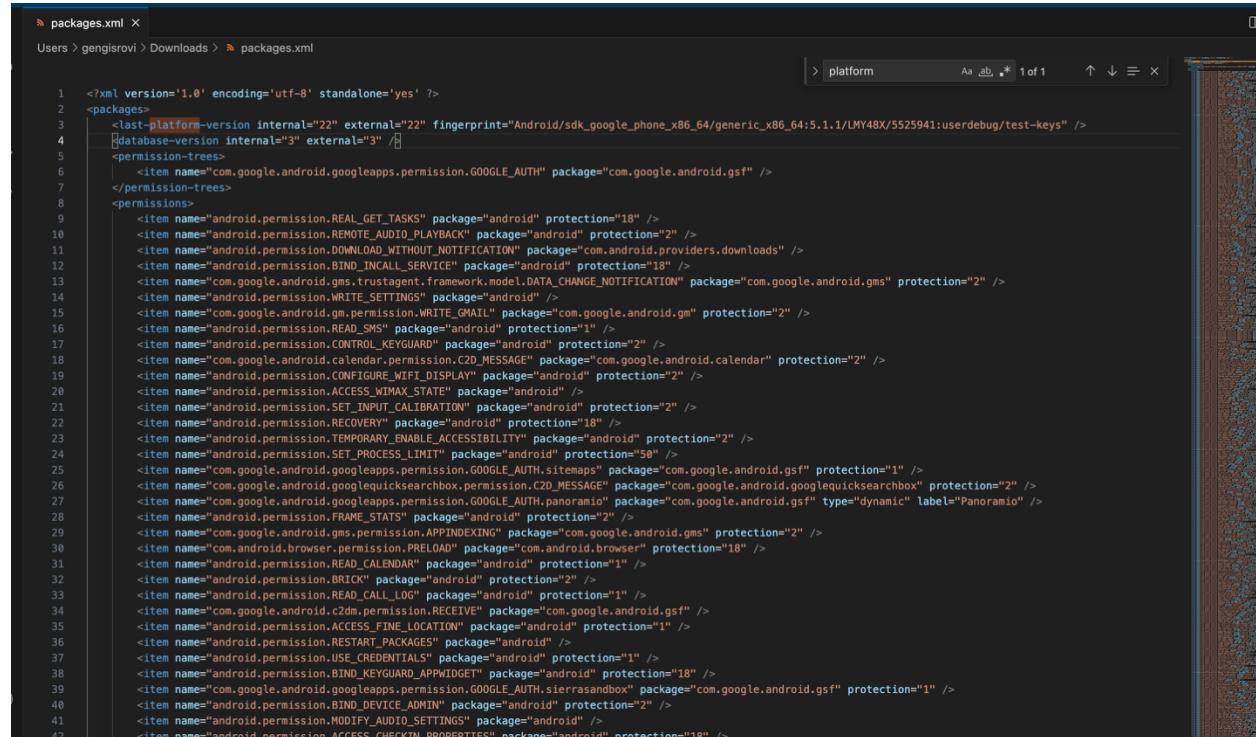
2.3.4. Información del Sistema y Software (Análisis de packages.xml)

Ante la ausencia del archivo build.prop en la partición de datos recuperada, se procedió al análisis forense del archivo de registro de paquetes packages.xml (Inodo recuperado mediante TSK), ubicado en /data/system/packages.xml. Este archivo contiene el inventario histórico de la instalación del sistema.

A. Versión del Sistema Operativo El análisis de la etiqueta XML <last-platform-version> reveló los siguientes metadatos:

- **SDK Version:** 22
- **Release:** Android 5.1.1
- **Fingerprint:** Android/sdk_google_phone_x86_64/generic_x86_64:5.1.1...

```
[gengisrovi@MacBookPro Downloads % fls -r data.dd | grep packages.xml
+ r/r 14307:    packages.xml
[gengisrovi@MacBookPro Downloads % icat data.dd 14307 > packages.xml
```



```
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<packages>
    <last-platform-version internal="22" external="22" fingerprint="Android/sdk_google_phone_x86_64/generic_x86_64:5.1.1/LMY48X/5525941:userdebug/test-keys" />
    <database-version internal="3" external="3" />
    <permission-trees>
        <item name="com.google.android.googleapps.permission.GOOGLE_AUTH" package="com.google.android.gsf" />
    </permission-trees>
    <permissions>
        <item name="android.permission.REAL_GET_TASKS" package="android" protection="18" />
        <item name="android.permission.REMOTE_AUDIO_PLAYBACK" package="android" protection="2" />
        <item name="android.permission.DOWNLOAD_WITHOUT_NOTIFICATION" package="com.android.providers.downloads" />
        <item name="android.permission.BIND_INCALL_SERVICE" package="android" protection="18" />
        <item name="com.google.android.gms.trustagent.framework.model.DATA_CHANGE_NOTIFICATION" package="com.google.android.gms" protection="2" />
        <item name="android.permission.WRITE_SETTINGS" package="android" />
        <item name="com.google.android.gm.permission.WRITE_GMAIL" package="com.google.android.gm" protection="2" />
        <item name="android.permission.READ_SMS" package="android" protection="1" />
        <item name="android.permission.CONTROL_KEYGUARD" package="android" protection="2" />
        <item name="com.google.android.calendar.permission.C2D_MESSAGE" package="com.google.android.calendar" protection="2" />
        <item name="android.permission.CONFIGURE_WIFI_DISPLAY" package="android" protection="2" />
        <item name="android.permission.ACCESS_WIMAX_STATE" package="android" />
        <item name="android.permission.SET_INPUT_CALIBRATION" package="android" protection="2" />
        <item name="android.permission.RECOVERY" package="android" protection="18" />
        <item name="android.permission.TEMPORARY_ENABLE_ACCESSIBILITY" package="android" protection="2" />
        <item name="android.permission.SET_PROCESS_LIMIT" package="android" protection="50" />
        <item name="com.google.android.googleapps.permission.GOOGLE_AUTH_sitemaps" package="com.google.android.gsf" protection="1" />
        <item name="com.google.android.googlequicksearchbox.permission.C2D_MESSAGE" package="com.google.android.googlequicksearchbox" protection="2" />
        <item name="com.google.android.googlequicksearchbox.permission.GOOGLE_AUTH_panoramio" package="com.google.android.gsf" type="dynamic" label="Panoramio" />
        <item name="android.permission.FRAME_STATS" package="android" protection="2" />
        <item name="com.google.android.gms.permission.APPINDEXING" package="com.google.android.gms" protection="2" />
        <item name="com.android.browser.permission.PRELOAD" package="com.android.browser" protection="18" />
        <item name="android.permission.READ_CALENDAR" package="android" protection="1" />
        <item name="android.permission.BRICK" package="android" protection="2" />
        <item name="android.permission.READ_CAL_LOG" package="android" protection="1" />
        <item name="com.google.android.c2dm.permission.RECEIVE" package="com.google.android.gsf" />
        <item name="android.permission.ACCESS_FINE_LOCATION" package="android" protection="1" />
        <item name="android.permission.RESTART_PACKAGES" package="android" />
        <item name="android.permission.USE_CREDENTIALS" package="android" protection="1" />
        <item name="android.permission.BIND_KEYGUARD_APPWIDGET" package="android" protection="18" />
        <item name="com.google.android.googleapps.permission.GOOGLE_AUTH_sierrasandboxx" package="com.google.android.gsf" protection="1" />
        <item name="android.permission.BIND_DEVICE_ADMIN" package="android" protection="2" />
        <item name="android.permission.MODIFY_AUDIO_SETTINGS" package="android" />
        <item name="android.permission.ACCESS_CHECKIN_PROPERTIES" package="android" protection="18" />
```

Figura 6. Evidencia de la versión del SO y tipo de dispositivo en packages.xml.

Hallazgo Forense: El dispositivo analizado no es un terminal físico convencional, sino un **Emulador Android x86_64**. El uso de un emulador es una técnica común en ciberdelincuencia para evitar dejar rastros físicos de hardware (IMEI real, ubicación GPS física) y para realizar

pruebas de penetración (hacking) en un entorno controlado. Por otro lado la cuenta registrada fue ceupe.forensics@gmail.com

B. Aplicaciones Instaladas (User Apps) Se filtraron las entradas del XML buscando la ruta /data/app/, la cual corresponde a aplicaciones instaladas por el usuario, descartando las aplicaciones de sistema (/system/app).

Se identificó una aplicación atípica instalada manualmente:

- **Nombre del Paquete:** org.troncoso.droidpond
 - **Ruta:** /data/app/org.troncoso.droidpond-1
 - **Permisos:** android.permission.INTERNET

```
<packages>
  <package name="com.android.inputdevices" codePath="/system/priv-app/InputDevices" nativeLibraryPath="/system/p...> "org.troncoso.droidpond Aa ab * 1 of 1 ↑ ↓ ≡ × 74
  ...
  <signing-keyset identifier="1" />
</package>
<package name="com.android.sdksetup" codePath="/system/app/SdkSetup" nativeLibraryPath="/system/app/SdkSetup/lib" flags="572997" ft="16a79d52d18" it="152279b2100" ut="16a...
  ...
  <sigs count="1">
    | <cert index="1" />
  </sigs>
  <proper-signing-keyset identifier="1" />
  <signing-keyset identifier="1" />
</package>
<package name="org.troncoso.droidpond" codePath="/data/app/org.troncoso.droidpond-1" nativeLibraryPath="/data/app/org.troncoso.droidpond-1/lib" flags="4767302" ft="15960...
  ...
  <sigs count="5">
    | <cert index="5" key="3082030d308201f5a00302010202046c0deb2d300d06092a864886f70d0101b05003037310b3009060355040613025533110300e60355040a1307416e64726f69643116301...
  </sigs>
  <perms>
    | <item name="android.permission.INTERNET" />
  </perms>
  <proper-signing-keyset identifier="17" />
  <signing-keyset identifier="17" />
</package>
<package name="com.google.android.apps.maps" codePath="/system/app/Maps" nativeLibraryPath="/system/app/Maps/lib" primaryCpuAbi="x86_64" flags="5783109" ft="16a79d54870...
  ...
  <sigs count="1">
    | <cert index="2" />
  </sigs>
  <proper-signing-keyset identifier="5" />
  <signing-keyset identifier="5" />
</package>
<package name="com.android.development_settings" codePath="/system/app/DevelopmentSettings" nativeLibraryPath="/system/app/DevelopmentSettings/lib" flags="572993" ft="16a...
  ...
  <sigs count="1">
```

Figura 7. Identificación de la aplicación objetivo instalada en el emulador.

Esta aplicación ([DroidPond](#)) no corresponde a software comercial estándar, lo que sugiere que es el software objetivo sobre el cual Mr. X estaba intentando realizar la ingeniería inversa o el ataque de fuerza bruta mencionado en los SMS

Conclusiones

La investigación ha permitido reconstruir la cadena delictiva completa. Las evidencias demuestran que Mr. X utilizó un **emulador Android versión 5.1.1** para ocultar su identidad física. En este entorno virtual, instaló la aplicación objetivo "**DroidPond**" (`org.troncoso.droidpond`), perteneciente a la organización atacada.

La trama consistía en vulnerar la seguridad de esta aplicación específica. Para ello:

1. Contactó inicialmente a **Matt** (650-555-1111) sin éxito.
 2. Contrató los servicios de **Danny Rand** (650-555-2222) por una suma de 10.000 (10k) semanales para labores de descifrado.

3. Obtuvo las herramientas de hacking (alojadas en el archivo `shadow`) a través de un enlace proporcionado por **Jessica (650-555-4444)**, al cual accedió desde el navegador nativo del emulador.

El análisis cruzado de los SMS (`mmssms.db`), el historial de navegación (`browser2.db`) y el registro de sistema (`packages.xml`) confirma de manera irrefutable la intencionalidad, los medios técnicos y los cómplices involucrados en el ciberataque.

3. DOCKER: INFRAESTRUCTURA Y DESPLIEGUE

En esta sección se resuelven los ejercicios de contenedorización, aplicando principios de seguridad como el principio de mínimo privilegio (usuarios no root) y persistencia de datos.

3.1. Ejercicio 1: Generador de ASCII Art

Se solicita una imagen Docker que convierta imágenes a ASCII art y guarde el resultado en un volumen persistente.

Código del Dockerfile: Se ha utilizado una imagen base ligera (`ubuntu:20.04`) y se ha creado un usuario específico para evitar la ejecución como `root`.

Dockerfile

```
FROM ubuntu:20.04
LABEL maintainer="Estudiante"
ENV DEBIAN_FRONTEND=noninteractive

# Instalación de jp2a y limpieza de caché para optimizar capas [cite: 1249]
RUN apt-get update && apt-get install -y jp2a \
    && rm -rf /var/lib/apt/lists/*

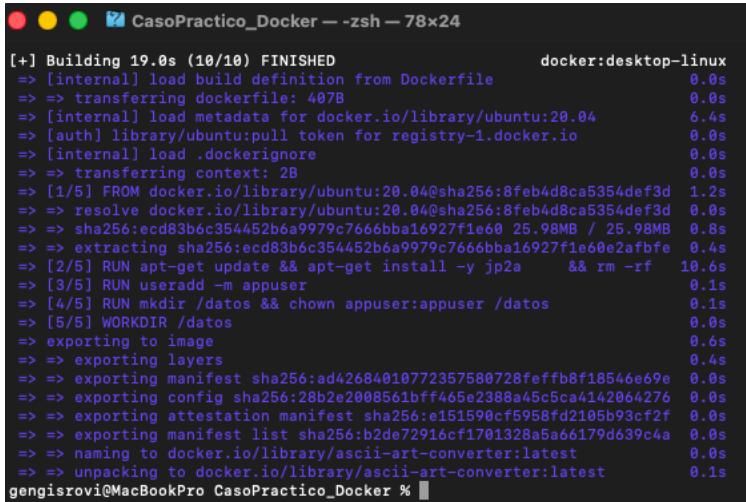
# Creación de usuario no-root por seguridad
RUN useradd -m appuser
RUN mkdir /datos && chown appuser:appuser /datos

# Definición del volumen persistente [cite: 1332]
VOLUME /datos

USER appuser
WORKDIR /datos
ENTRYPOINT ["jp2a"]
```

Proceso de Construcción y Ejecución:

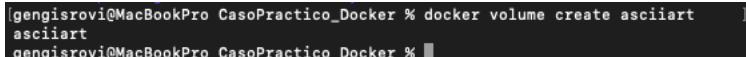
1. Construcción de la imagen: docker build -t ascii-art-converter .



```
[+] Building 19.0s (10/10) FINISHED
--> [internal] load build definition from Dockerfile          docker:desktop-linux
--> => transferring dockerfile: 407B                         0.0s
--> [internal] load metadata for docker.io/library/ubuntu:20.04      0.4s
--> [auth] library/ubuntu:pull token for registry-1.docker.io      0.0s
--> [internal] load .dockerignore                                0.0s
--> => transferring context: 2B                                0.0s
--> [1/5] FROM docker.io/library/ubuntu:20.04@sha256:8feb4d8ca5354def3d 1.2s
--> => resolve docker.io/library/ubuntu:20.04@sha256:8feb4d8ca5354def3d 0.0s
--> => sha256:ecdb83b6c354452b6a9979c7666bba16927f1e60 25.98MB / 25.98MB 0.8s
--> => extracting sha256:ecdb83b6c354452b6a9979c7666bba16927f1e60e2fbfe 0.4s
--> [2/5] RUN apt-get update && apt-get install -y jp2a    && rm -rf 10.6s
--> [3/5] RUN useradd -m appuser                           0.1s
--> [4/5] RUN mkdir /datos && chown appuser:appuser /datos 0.1s
--> [5/5] WORKDIR /datos                                 0.0s
--> exporting to image                                    0.6s
--> => exporting layers                                  0.4s
--> => exporting manifest sha256:ad42684010772357580728feffb8f18546e69e 0.0s
--> => exporting config sha256:28b2e2008561bfff465e2388a45c5ca4142064276 0.0s
--> => exporting attestation manifest sha256:e15159bcf958fd2105b93cf2f 0.0s
--> => exporting manifest list sha256:b2de72916cf1701328a5a66179d639c4a 0.0s
--> => naming to docker.io/library/ascii-art-converter:latest 0.0s
--> => unpacking to docker.io/library/ascii-art-converter:latest 0.1s
gengisrovi@MacBookPro CasoPractico_Docker %
```

Figura 7. Construcción de la imagen Docker.

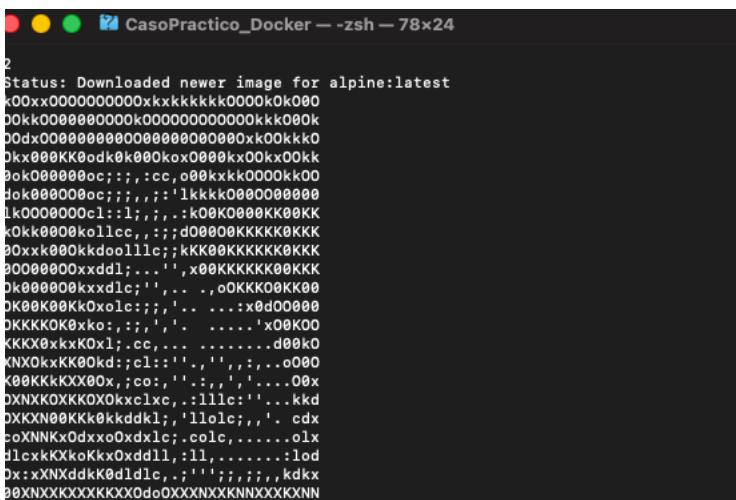
2. Creación del volumen: docker volume create asciaiart



```
[gengisrovi@MacBookPro CasoPractico_Docker % docker volume create asciaiart      ]
[gengisrovi@MacBookPro CasoPractico_Docker %]
```

Figura 8. Creación del volumen persistente.

3. Ejecución y Prueba: Se ejecutó el contenedor mapeando una imagen local (`test.jpg`) y guardando la salida en el volumen. `docker run --rm -e TERM=xterm -v $(pwd) :/entrada -v asciaiart:/datos ascii-art-converter --output=/datos/resultado.txt /entrada/test.jpg`



```
2
Status: Downloaded newer image for alpine:latest
<00xx0000000000xkkkkkk0000k0k080
00kk00000000k000000000000kk000k
00d000000000000000000000000x00kkk0
0kx000K00dk0k000kox000kx00k00kk
0ok000000c:::,cc,0@0xkk00000k00
dok000008oc::,::,'lkkkk0000000000
1k0000000cl::1;,:..:k00K0000KK00KK
<0kk0000kolcc,,,:d00000KKKK0KKK
00xxk000Kkdoolllc;kkk00KKKKKK0KKK
00000000xxddl;...'',x00KKKKKK0KKK
0k000000kxdlc;''...,o0KKK08KK00
0K00K00K0K0xlc:;,'.....:0d00000
0KKKK0K0xko:,:,:'. ....'x00K00
KKKx0kxK0x1;.cc,.....d00K0
KX0kxKK00kd:;cl:,'.,',;,...o00
K00KKKKX00x;,co:,'.,',;,...08x
DXNXK0XKK0X0kxlxc,.lllc:;'..kkd
DXKXN00KKk0kkddk1;,'llolc;,'. cdx
coXNKx0dxo0xdxl;c,colc,....olx
dlcxxxKk0Kx0xdll,:ll,.....:lod
Dx:xxNxddkK0d1d1c,.;''';,;,;,kdkx
00XNXXKXXXXXX0do0XXXNXXKNNNNNNXNN
```

Figura 9. Ejecución exitosa y generación del fichero en el volumen.

3.2. Ejercicio 2: Arquitectura WordPress + MySQL

Se despliega una arquitectura web con base de datos, garantizando la persistencia mediante volúmenes y la comunicación mediante una red interna.

3.2.1. Despliegue mediante Docker Compose

Se ha optado por Docker Compose para la orquestación, ya que permite definir la infraestructura como código de manera clara y reproducible.

Fichero docker-compose.yml:

```
YAML
version: '3.8'

services:
  db:
    image: mysql:5.7
    container_name: mysql_db
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: password_root
      MYSQL_DATABASE: wordpress
      MYSQL_USER: wp_user
      MYSQL_PASSWORD: wp_password
    volumes:
      - db_data:/var/lib/mysql
    networks:
      - wp-net

  wordpress:
    image: wordpress:latest
    container_name: wordpress_app
    restart: always
    depends_on:
      - db
    ports:
      - "8080:80"
    environment:
      WORDPRESS_DB_HOST: db
      WORDPRESS_DB_USER: wp_user
      WORDPRESS_DB_PASSWORD: wp_password
      WORDPRESS_DB_NAME: wordpress
    volumes:
      - wp_data:/var/www/html
    networks:
      - wp-net

volumes:
  db_data:
  wp_data:

networks:
  wp-net:
```

```
driver: bridge
```

Despliegue y Verificación:

1. Ejecución del entorno: docker-compose up -d

```
[gengisrovi@MacBookPro CasoPractico_Docker % docker-compose up -d      ]
[+] Running 37/37
  ✓ wordpress Pulled
  ✓ db Pulled
[+] Running 5/5
  ✓ Network casopractico_docker_wp-net  Created          11.6s
  ✓ Volume casopractico_docker_db_data  Created          9.9s
  ✓ Volume casopractico_docker_wp_data  Created          0.0s
  ✓ Container mysql_db                Started          0.0s
  ✓ Container wordpress_app           Started          0.7s
  ✓ Container wordpress_app           Started          0.3s
gengisrovi@MacBookPro CasoPractico_Docker %
```

Figura 10. Despliegue de la pila de servicios con Docker Compose.

2. Estado de los contenedores: docker-compose ps

```
[gengisrovi@MacBookPro CasoPractico_Docker % docker-compose ps      ]
NAME          IMAGE           COMMAND          SERVICE    CREATE
D STATUS       PORTS
mysql_db      mysql:5.7    "docker-entrypoint.s..."  db        4 minu
tes ago      Up 4 minutes  3306/tcp, 33060/tcp
wordpress_app wordpress:latest "docker-entrypoint.s..."  wordpress 4 minu
tes ago      Up 4 minutes  0.0.0.0:8080->80/tcp, [::]:8080->80/tcp
gengisrovi@MacBookPro CasoPractico_Docker %
```

Figura 11. Verificación de contenedores activos.

3. Comprobación funcional: Acceso a través del navegador a <http://localhost:8080>.

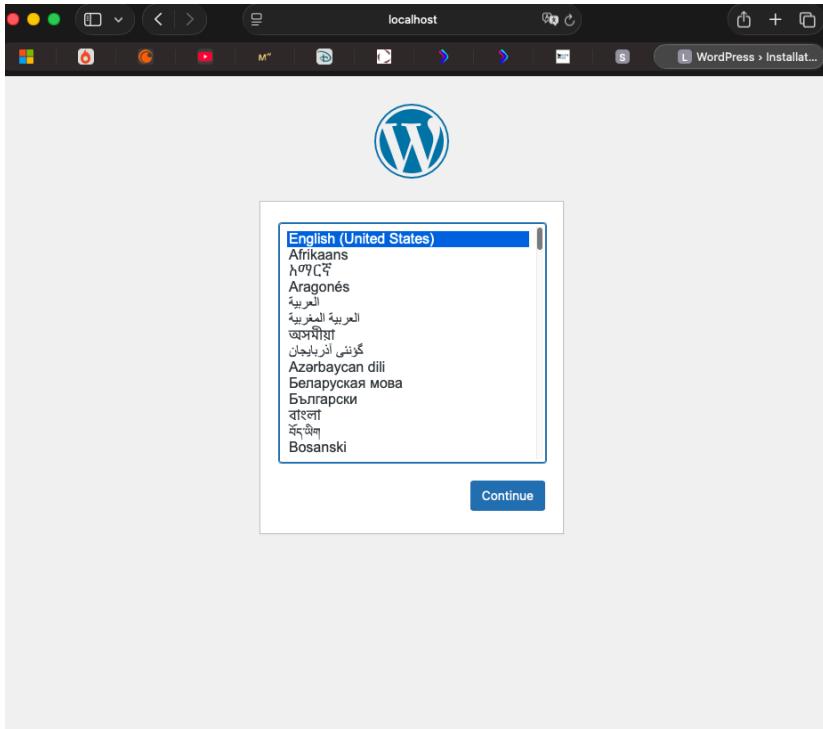


Figura 12. Acceso exitoso a la aplicación WordPress.

4. CONCLUSIONES

A través del desarrollo de este caso práctico se han alcanzado las siguientes conclusiones:

1. **Análisis Forense:** La extracción y análisis de bases de datos SQLite específicas (`mmsms.db`, `contacts2.db`, `browser2.db`) alojadas en las particiones de datos del sistema Android es fundamental para reconstruir la cronología de los hechos delictivos. El uso de herramientas como *The Sleuth Kit* permitió acceder a la información sin comprometer la integridad de la imagen original.
2. **Seguridad en Docker:** La implementación de contenedores debe seguir principios de seguridad desde el diseño. En el ejercicio 1, se demostró cómo evitar la ejecución como `root` mediante la directiva `USER` en el Dockerfile.
3. **Orquestación y Persistencia:** El uso de Docker Compose simplifica drásticamente el despliegue de arquitecturas multicontenedor (WordPress + MySQL), gestionando automáticamente la red interna para la comunicación segura entre servicios y los volúmenes para garantizar que la información no se pierda al reiniciar los contenedores.