



Análisis Forense Avanzado

Laboratorio Práctica 1

Paula Daniela Sánchez Rodríguez

Enrique García Cuadrado

2025



Contenido

1. Informe forense del móvil de Jeff Bezos	3
Cronología de los hechos.....	3
Herramientas Hardware y Software	4
Investigue sobre la herramienta PowerGREP.	5
Explique el procedimiento seguido por los analistas	6
¿Cómo podría hacer la verificación del tráfico generado por su móvil? ¿Sería necesaria una infraestructura costosa?	8
¿Por qué cree que se usa un entorno Sandbox para capturar el tráfico saliente?.....	9
Explique el apartado referente a los IOCs (Indicadores de compromiso) que presenta el informe. ¿Qué le parecen las URLs que se muestran? ¿Son maliciosas?	9
2. Evidencias en Linux	10
Investigue sobre la siguiente orden y sus parámetros.....	10
<i>dd if=/dev/nombre_interno_USB_origen of=/dev/nombre</i>	10
Investigue el uso de las siguientes herramientas ampliamente utilizadas	10
2.1 Adquisición de evidencias Linux	11
Cree un fichero llamado particiones.txt con la lista de particiones	11
Cree otro fichero llamado hash_particiones.txt con la huella digital generada.....	12
Abra el navegador Firefox y descargue varias imágenes de una página Web	12
Genere otro fichero comprobación.txt con el hash del fichero huellas_imgs.txt	12
Usando strings genere un archivo llamado versión.txt con.....	12
Genere otro archivo llamado hash_versión.txt con la huella del fichero versión.txt.....	13
Genere un archivo llamado hash_version2.txt con la huella.....	13
Compruebe que ambas huellas digitales son diferentes.....	13
Utilizando el comando grep, muestre por pantalla las líneas	13
Genere un fichero llamado evm.txt	14
Genere su huella digital en un fichero llamado huella_evm.txt.....	15
Compruebe el funcionamiento de la orden wc	15
3. Actividades opcionales	16
Bloqueos de rangos de IP de Cloudflare por parte de Movistar	16
Evidencias de conexiones a internet para app de DNI.....	17

1. Informe forense del móvil de Jeff Bezos

Lo más interesante de este ataque es que el usuario, Jeff Bezos, no tuvo que pinchar ningún enlace ni descargar ningún archivo. El ataque se basaba en una vulnerabilidad zero-day que explotaba la descarga automática de contenidos en WhatsApp, y aunque el vídeo que le enviaron tenía un tamaño muy grande lo que podría indicar la presencia de malware, fue un proceso automático y no tuvo nada que hacer.

Cronología de los hechos

4 de abril de 2018 Jeff Bezos y el príncipe heredero de Arabia Saudí Mohammed Bin Salman se intercambian los números de teléfono durante una cena en Los Ángeles e intercambian dos mensajes a través WhatsApp.

1 de mayo de 2018 Jeff Bezos recibe un mensaje de WhatsApp desde la cuenta utilizada por Mohammed. Este mensaje contenía un video adjunto de 4,22 MB enviado sin previo aviso ni discusión.

2 de mayo de 2018 Se registran 126 MB de datos exfiltrados desde el iPhone de Jeff Bezos mientras que el día anterior habían sido solo 430KB. Este comportamiento continuó durante meses.

8 de noviembre de 2018 Mohammed vuelve a enviar un mensaje a Jeff Bezos con información privada, una foto de una mujer que se parecía a Lauren Sánchez, su amante en ese momento y un mensaje, ya que esta relación no era pública.

14 de febrero de 2019 Jeff Bezos recibe otro mensaje por WhatsApp con un informe sobre la campaña que está haciendo en internet Arabia Saudí en contra suya.

16 de febrero de 2019 Mohammed envía un mensaje de WhatsApp a Bezos negando cualquier acción en su contra o en contra de Amazon.

17 de febrero de 2019 El asesor de seguridad de Jeff Bezos, Gavin de Becker, recibe información sobre una posible amenaza persistente avanzada (APT) dirigida al teléfono de Jeff y recomienda un análisis forense del dispositivo.

24 de febrero de 2019 Gavin de Becker, contrata a Anthony Ferrante de FTI consulting, para realizar un análisis forense del iPhone X de Jeff Bezos.

25 de febrero de 2019 FTI inicia un primer análisis de un archivo adjunto sospechoso en el teléfono de Bezos.

17 de mayo de 2019 FTI establece un laboratorio forense para analizar el iPhone. Se realiza una adquisición forense del dispositivo y se analizan los artefactos digitales.

18 de mayo de 2019 se recibe el dispositivo y se asegura el laboratorio para el análisis.

19 de mayo de 2019 FTI realiza un escaneo de malware en el iPhone pero no se detecta nada. Se inicia la captura de tráfico de red desde el iPhone para analizar el comportamiento del dispositivo.

22 de mayo de 2019 FTI devuelve el iPhone a Gavin y continúa analizando los datos forenses recopilados.

Mayo – julio de 2019 se realiza un análisis exhaustivo de las evidencias forenses.

19 de Julio de 2019 FTI concluye que el iPhone de Jeff Bezos fue comprometido a través del archivo de video enviado por WhatsApp desde la cuenta de Mohammed y se observa que la exfiltración de datos continuó durante meses.

Noviembre de 2019 se publica el informe pericial realizado por FTI Consulting confirmando que el hackeo probablemente fue facilitado por herramientas adquiridas por Saud al Qahtani, asesor de Mohammed y jefe de ciberseguridad de Arabia Saudí.

Herramientas Hardware y Software que se han utilizado en el análisis forense

Herramientas de Hardware

1. Microsoft Surface (2 unidades)
 - Especificaciones:
 - Memoria: 16 GB
 - Procesador: Intel Core i7
 - Almacenamiento: 512 GB
 - Estos ordenadores se utilizaron para ejecutar el software forense y realizar el análisis de los datos extraídos del iPhone.
2. Router inalámbrico NETGEAR R6700
 - Configuración:
 - ESSID oculto.
 - Cifrado WPA2 de 128 bits.
 - Red interna sin conexión a Internet.
 - Este router se configuró para crear una red interna segura dentro del laboratorio forense. Al no tener conexión a Internet se evitó cualquier riesgo de contaminación externa o fuga de datos durante el análisis.

Herramientas de Software

1. Cellebrite UFED 4PC Ultimate

Esta herramienta permite el análisis forense de dispositivos móviles incluyendo la extracción de datos del sistema de archivos, contraseñas, y la clonación de

tarjetas SIM. En este caso, se utilizó para obtener una imagen forense del iPhone X de Jeff Bezos.

2. Cellebrite Physical Analyzer

Esta herramienta se utiliza para analizar la imagen forense obtenida, la cual permite examinar archivos, mensajes, llamadas, y otros datos del dispositivo en busca de evidencias o actividad maliciosa.

3. Telerik Fiddler

Fiddler se configuró como un proxy man-in-the-middle para capturar y analizar el tráfico de red generado por el iPhone X en un sandbox. Esto permitió simular una conexión a internet y monitorear las comunicaciones del dispositivo en busca de actividad sospechosa.

4. Wireshark

Es una herramienta de análisis de paquetes de red que se utilizó para capturar y examinar el tráfico de red en tiempo real. Junto con Fiddler, permitió identificar posibles comunicaciones como la exfiltración de datos.

5. Oracle VirtualBox con Ubuntu 18.04 LTS

Se utilizaron máquinas virtuales con Ubuntu para ejecutar herramientas de análisis de malware y procesamiento de datos. Estas permiten realizar análisis sin afectar el sistema operativo anfitrión y facilitan la restauración a un estado anterior en caso de problemas con el uso de snapshots.

6. PowerGREP

Es una herramienta de procesamiento de datos y sobre todo de búsqueda de patrones. Se utilizó para buscar y correlacionar datos extraídos del iPhone con una base de datos de inteligencia de amenazas con el fin de identificar posibles indicadores de compromiso.

7. Windows Subsystem for Linux con Ubuntu 18.04 LTS

Esto dejó ejecutar utilidades de Linux en un entorno de Windows facilitando el análisis de los datos y la colaboración entre los dos ordenadores utilizados en el laboratorio.

Investigue sobre la herramienta PowerGREP e indique un posible ejemplo de uso.

PowerGREP es una herramienta para buscar, extraer y manipular texto en archivos y documentos. Es muy utilizada en tareas de procesamiento de datos, análisis forense y administración de sistemas, especialmente cuando se tiene que trabajar con grandes volúmenes de datos o realizar búsquedas complejas en múltiples archivos. Algunas de sus características principales incluyen:

- Realizar búsquedas utilizando expresiones regulares, lo que facilita la localización de patrones específicos en textos o información específica de archivos, como direcciones de correo electrónico, números de teléfono, URLs, etc.
- Correlacionar datos extraídos con bases de datos externas, lo que es muy útil en análisis forenses y automatizar tareas repetitivas mediante scripts.

Explique el procedimiento seguido por los analistas para identificar el aumento de tráfico saliente del móvil hackeado.

1. Configuración de la red aislada

- Se configuró un router NETGEAR R6700 Nighthawk AC1750 con una red interna sin conexión a Internet y se utilizó WPA2 de 128 bits para cifrar la red además de que se ocultó el ESSID para mayor seguridad.
- Esta configuración se utilizó para simular una conexión a Internet en un entorno controlado y seguro.

2. Preparación de las estaciones de trabajo:

- Se utilizaron dos ordenadores Microsoft Surface con Cellebrite UFED 4PC Ultimate y Physical Analyzer instalados.
- Se configuró Telerik Fiddler como un proxy man-in-the-middle para capturar el tráfico de red.
- Se instaló Wireshark para capturar y analizar paquetes de red en tiempo real.

3. Adquisición forense del iPhone X

- Se realizó una imagen bit a bit del iPhone X utilizando Cellebrite.
- Se descubrió que el dispositivo tenía cifrado de copia de seguridad de iTunes activado lo que impedía un análisis completo inicialmente. Para evitar el cifrado se restablecieron todas las configuraciones del iPhone X a los valores de fábrica, lo que eliminó la contraseña de cifrado sin afectar los datos del usuario.
- Después del reseteo se realizó una imagen forense completa del dispositivo.

4. Captura y análisis del tráfico de red

- Configuración del proxy y captura de paquetes:
 - Se configuró el iPhone X para redirigir todo su tráfico Wi-Fi a través del proxy Fiddler.
 - Se utilizó Wireshark para capturar los paquetes de red generados por el dispositivo.

- Simulación de actividad del usuario:
 - Se realizaron varias pruebas para capturar el tráfico de red en diferentes estados del dispositivo:
 - Dispositivo bloqueado.
 - Dispositivo desbloqueado.
 - Dispositivo inactivo.
 - Simulación de actividad del usuario (abrir y cerrar aplicaciones).
- Captura de tráfico durante varios días:
 - La captura de tráfico de red se realizó desde el 19 de mayo de 2019 hasta el 21 de mayo de 2019, lo que permitió obtener una muestra significativa del comportamiento del dispositivo.

5. Análisis de los datos de tráfico

- Línea de base del tráfico normal:
 - Se estableció una línea de base del tráfico saliente normal del iPhone X antes del 1 de mayo de 2018 (fecha en que se recibió el video sospechoso).
- Identificación de picos de tráfico:
 - Después de la línea de base, se observó el aumento en el tráfico saliente
- Comparación con dispositivos de control:
- Se comparó el tráfico saliente del iPhone X con el de cinco iPhones de control propiedad de FTI.

6. Correlación con eventos externos

- El aumento del tráfico saliente coincidió con la recepción del archivo de video de WhatsApp el 1 de mayo de 2018.
- También se observaron mensajes de WhatsApp enviados desde la cuenta de Mohammed que parecían hacer referencia a información privada de Bezos, lo que sugiere que el dispositivo estaba siendo monitoreado.

7. Conclusiones del análisis

- FTI concluyó con un nivel de confianza medio a alto que el iPhone X de Jeff Bezos fue comprometido a través del archivo de vídeo enviado por WhatsApp.
- El aumento masivo del tráfico saliente, junto con la falta de explicaciones legítimas para este comportamiento, respaldó la conclusión de que el dispositivo estaba siendo utilizado para exfiltrar datos de manera no autorizada.

Desde el punto de vista del laboratorio y teniendo en cuenta la información del informe ¿cómo podría hacer la verificación del tráfico generado por su móvil? ¿Sería necesaria una infraestructura costosa?

Para replicar el análisis realizado en el informe de FTI Consulting y verificar el tráfico generado por un móvil, se pueden seguir los siguientes pasos:

1. Configuración del entorno de laboratorio:

- El teléfono debe ser aislado en un entorno controlado para evitar interferencias externas y garantizar que el tráfico generado sea únicamente el del dispositivo bajo análisis.
- Utilizar una red sandbox (aislada) para simular una conexión a Internet sin exponer el dispositivo a la red real. Esto permite capturar y analizar el tráfico de red de manera segura.

2. Captura del tráfico de red:

- Se puede utilizar herramientas como Wireshark para capturar el tráfico de red en tiempo real.
- Se configura un proxy para actuar como intermediario entre el teléfono y la red. Esto permite inspeccionar y registrar todas las solicitudes y respuestas de red.

3. Análisis del tráfico capturado:

- Se buscan patrones de tráfico inusuales, como conexiones a servidores desconocidos o en países sospechosos, grandes volúmenes de datos enviados o comunicaciones en momentos inusuales.
- Los datos capturados se comparan con bases de datos de inteligencia sobre amenazas para identificar si alguna de las conexiones está asociada con actividades maliciosas.

4. Análisis forense del dispositivo:

- Se crea una imagen forense del teléfono utilizando herramientas para obtener una copia exacta del sistema de archivos.
- Se analiza la imagen forense en busca de indicios de malware, como archivos sospechosos, aplicaciones no autorizadas o modificaciones en el sistema.

5. Validación de los hallazgos:

- Se intenta reproducir el tráfico sospechoso en un entorno controlado para confirmar si está relacionado con actividades maliciosas.
- Se observa el comportamiento del teléfono en diferentes condiciones (bloqueado, desbloqueado, en uso, en reposo) para identificar anomalías.

No es necesaria una infraestructura extremadamente costosa para verificar el tráfico generado por un móvil, pero sí se requiere un conjunto de herramientas especializadas y un entorno controlado como lo tenían en ese laboratorio. La inversión principal estaría

en el hardware (ordenadores con cierta potencia, routers, etc.) y en el software profesional (como Cellebrite), aunque existen alternativas de código abierto gratis

¿Por qué cree que se usa un entorno Sandbox para capturar el tráfico saliente?

El uso de una sandbox para capturar el tráfico saliente es para poder analizar el comportamiento del archivo malicioso sin afectar otros sistemas y sin tener ruido.

De esta forma se puede ejecutar el archivo de malware en un entorno controlado, evitando que comprometa sistemas de producción o activos críticos y analizar todo el tráfico de red generado por el archivo, identificando conexiones a servidores de comando y control, direcciones IP sospechosas o la exfiltración de datos que es lo que se vio en este caso.

Si el ataque utiliza exploits, el sandbox permite interceptar esta actividad y descifrar datos sin riesgo para el sistema operativo principal, dándonos así el conocimiento o la inteligencia del atacante sin correr riesgos.

En este caso la captura del tráfico que se hizo dentro de la sandbox permitió identificar el flujo de datos inusuales y los posibles servidores a los que se enviaba la información, lo que ayudó a reconstruir el ataque.

Explique el apartado referente a los IOCs (Indicadores de compromiso) que presenta el informe. ¿Qué le parecen las URLs que se muestran? ¿Son maliciosas?

En el reporte del hackeo al teléfono de Jeff Bezos, el apartado sobre indicadores de compromiso da detalles sobre las cosas que sugieren que ocurrió una intrusión. Los IOCs incluyen URLs sospechosas, direcciones IP, hashes de archivos y otras evidencias forenses que se identificaron durante la investigación.

Se mencionan URLs específicas asociadas con el ataque, que fueron utilizadas para la entrega del malware o la exfiltración de datos y se identifican direcciones IP relacionadas con los servidores desde los cuales se originó la actividad maliciosa.

Estas URLs del informe muestran actividad sospechosa, ya que están vinculadas al tráfico saliente del dispositivo de Jeff Bezos después de recibir el archivo de WhatsApp infectado. Específicamente se detectó una conexión con un dominio que no formaba parte de las comunicaciones legítimas esperadas y están relacionadas con servidores posiblemente usados para C2 comando y control, algo muy común que hacen hoy en día los grupos APT.

2. Evidencias en Linux

Investigue sobre la siguiente orden y sus parámetros

```
dd if=/dev/nombre_interno_USB_origen of=/dev/nombre_interno_USB_destino bs=64K conv=noerror,sync
```

dd → es una herramienta de Linux utilizada para copiar datos a nivel de bloque de un dispositivo o archivo a otro.

if=/dev/nombre_interno_USB_origen → Especifica el archivo de entrada (if = "input file"), en este caso, el dispositivo de almacenamiento de origen (una unidad USB, por ejemplo). Se debe reemplazar 'nombre_interno_USB_origen' por el nombre del dispositivo real, por ejemplo /dev/sdb.

of=/dev/nombre_interno_USB_destino → Especifica el archivo de salida (of = "output file"), es decir la unidad de destino donde se clonará el contenido del dispositivo de origen.

bs=64K → Define el tamaño del bloque de lectura y escritura en 64 KB. Esto permite que la transferencia de datos sea más eficiente en comparación con el valor predeterminado

conv=noerror,sync → Modifica la forma en que dd maneja errores y sincroniza datos:

- **noerror** → Indica que dd debe continuar copiando incluso si encuentra errores de lectura en el dispositivo de origen, en lugar de detenerse.
- **sync** → Asegura que si hay errores de lectura los bloques defectuosos se rellenarán con ceros para mantener la alineación de los datos y evitar un desfase en la imagen resultante.

Investigue el uso de las siguientes herramientas ampliamente utilizadas en análisis forense y que podemos ejecutar en Linux, para usarlas en el siguiente apartado : a) fdisk b) netcat c) sha256sum d) strings e) dmesg

FDISK es una herramienta para la gestión de particiones en sistemas Linux. Su uso en análisis forense se enfoca en la capacidad de listar las particiones de un disco y verificar su estructura sin modificarla. Con fdisk -l podemos tener una visión general de la distribución del almacenamiento, identificar particiones ocultas y preparar la adquisición de imágenes forenses de discos completos.

NETCAT es una herramienta muy versátil para la manipulación de conexiones de red. En forense se puede usar para transferir imágenes de discos completos entre máquinas sin alterar su contenido. También es útil para establecer conexiones remotas y capturar tráfico de red permitiendo a los analistas interceptar y analizar datos en tránsito.

SHA256SUM sirve para verificar la integridad de archivos y evidencias digitales. En un análisis forense se usa para calcular la huella digital o hash de archivos y comparar estos valores antes y después de su análisis. Esto garantiza que la evidencia digital no

ha sido alterada en el proceso, manteniendo la cadena de custodia y asegurando que los datos analizados son exactamente iguales a los de su estado original.

STRINGS deja extraer cadenas de texto imprimibles de archivos binarios. Ayuda a revelar información oculta dentro de archivos ejecutables, volúmenes de memoria o imágenes forenses. Al ejecutarlo sobre un archivo sospechoso se pueden descubrir fragmentos de texto incrustados que pueden contener contraseñas, direcciones de red, comandos o incluso mensajes encubiertos.

DMESG da acceso a los registros del kernel, mostrando eventos de hardware y procesos del sistema. Esta utilidad es clave en la investigación de incidentes, ya que permite revisar logs del sistema en busca de actividad sospechosa, como la conexión de dispositivos USB no autorizados, errores de memoria o actividad anómala en los módulos del kernel. Al filtrar los registros con `grep`, los analistas pueden enfocarse en eventos específicos y reconstruir la línea de tiempo de un incidente.

2.1 Adquisición de evidencias Linux

Cree un fichero llamado `particiones.txt` con la lista de particiones del sistema usando `fdisk`.

```
sansforensics@siftworkstation: ~
$ sudo fdisk -l > particiones.txt
sansforensics@siftworkstation: ~
$ ls
Desktop Documents Downloads examples.desktop Music particiones.txt Pictures Public Templates Videos
sansforensics@siftworkstation: ~
$ cat particiones.txt
Disk /dev/loop1: 14.8 MiB, 15462400 bytes, 30200 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop2: 44.9 MiB, 47063040 bytes, 91920 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop3: 160.2 MiB, 167931904 bytes, 327992 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop6: 54.7 MiB, 57294848 bytes, 111904 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

`sudo fdisk -l`: Lista todas las particiones del sistema.

`>`: Redirige la salida del comando a un archivo.

`particiones.txt`: Nombre del archivo donde se almacenará la información.

Cree otro fichero llamado hash_particiones.txt con la huella digital generada con sha256sum del fichero particiones.txt.

```
sansforensics@siftworkstation: ~
$ sha256sum particiones.txt > hash_particiones.txt
sansforensics@siftworkstation: ~
$ ls
Desktop Documents Downloads examples.desktop hash_particiones.txt Music particiones.txt Pictures Public Templates Videos
sansforensics@siftworkstation: ~
$ cat hash_particiones.txt
0aec8e628c0396879dd699501cf3899b25aef6b63c45a9066c2f748e60d40378  particiones.txt
```

sha256sum particiones.txt: calcula la huella digital (hash) SHA-256 del archivo particiones.txt.

>: redirige la salida del comando al archivo hash_particiones.txt.

hash_particiones.txt: archivo donde se almacena el hash generado.

Abra el navegador Firefox y descargue varias imágenes de una página Web. Utilizando sha256sum cree un fichero llamado huellas_imgs.txt con los hashes de estas imágenes.

```
sansforensics@siftworkstation: ~
$ sha256sum Downloads/1024_6822.jpeg > huellas_imgs.txt
sansforensics@siftworkstation: ~
$ sha256sum Downloads/1024_682.jpeg > huellas_imgs.txt
bash: huellas_imgs.txt: cannot overwrite existing file
sansforensics@siftworkstation: ~
$ sha256sum Downloads/1024_682.jpeg >> huellas_imgs.txt
sansforensics@siftworkstation: ~
$ sha256sum Downloads/1200_800.jpeg >> huellas_imgs.txt
sansforensics@siftworkstation: ~
$ cat huellas_imgs.txt
c43fb7a890bf86049508e7a4ed0be37f6e2153e49457b7efff48aabc65d689e Downloads/1024_6822.jpeg
749f13f9c8dda711feedc98cf839b3c8a282489d478aa586957e8f43f9b56177 Downloads/1024_682.jpeg
09716039fabd3ba17dbb42575ae56e7a87c3b6f7ed2169f7f69c3fa1f6bd8571 Downloads/1200_800.jpeg
sansforensics@siftworkstation: ~
```

Para no sobrescribir el archivo de huellas se usa un doble >>....

Genere otro fichero comprobación.txt con el hash del fichero huellas_imgs.txt

```
sansforensics@siftworkstation: ~
$ sha256sum huellas_imgs.txt > comprobacion.txt
sansforensics@siftworkstation: ~
$ cat comprobacion.txt
b007f1eab6c9647295f7d2128c840d55ed2cc8b41104f80b10aca0a300084c7c  huellas_imgs.txt
sansforensics@siftworkstation: ~
```

Usando strings genere un archivo llamado versión.txt con el número de versión de esta herramienta utilizando uno de sus parámetros.

```
b007f1eab6c9647295f7d2128c840d55ed2cc8b41104f80b10aca0a300084c7c  huellas_imgs.txt
sansforensics@siftworkstation: ~
$ strings --version > version.txt
sansforensics@siftworkstation: ~
$ cat version.txt
GNU strings (GNU Binutils for Ubuntu) 2.30
Copyright (C) 2018 Free Software Foundation, Inc.
This program is free software; you may redistribute it under the terms of
the GNU General Public License version 3 or (at your option) any later version.
This program has absolutely no warranty.
sansforensics@siftworkstation: ~
```

Genere otro archivo llamado hash_versión.txt con la huella del fichero versión.txt

```
sansforensics@siftworkstation: ~  
$ sha256sum version.txt > hash_version.txt  
sansforensics@siftworkstation: ~  
$ cat hash_version.txt  
ad23d785549ada842d99921129e6afbcc2302a040b38243d7ffbee839e6bccdc version.txt  
sansforensics@siftworkstation: ~
```

Modificamos con el editor nano el fichero llamado versión.txt introduciendo cambios y caracteres aleatorios.

```
File Edit View Search Terminal Help  
GNU nano 2.9.3 version.txt  
  
GNU strings (GNU Binutils for Ubuntu) 2.30  
Copyright (C) 2018 Free Software Foundation, Inc.  
This program is free software; you may redistribute it under the terms of  
the GNU General Public License version 3 or (at your option) any later version.  
This program has absolutely no warranty.
```

Genere un archivo llamado hash_version2.txt con la huella del fichero modificado versión.txt

```
sansforensics@siftworkstation: ~  
$ sha256sum version.txt > hash_version2.txt  
sansforensics@siftworkstation: ~  
$ cat hash_version2.txt  
14012c9511162d6f751e35d1ec120c76cb1a1a91f5990c826e84644c3d190caa version.txt  
sansforensics@siftworkstation: ~
```

Compruebe que ambas huellas digitales son diferentes.

```
sansforensics@siftworkstation: ~  
$ diff hash_version.txt hash_version2.txt  
1c1  
< ad23d785549ada842d99921129e6afbcc2302a040b38243d7ffbee839e6bccdc version.txt  
---  
> 14012c9511162d6f751e35d1ec120c76cb1a1a91f5990c826e84644c3d190caa version.txt  
sansforensics@siftworkstation: ~
```

Como se observa ambos hashes son completamente diferentes, lo que indica que el hash del archivo versión.txt ha cambiado lo que en análisis forense indicaría que el archivo ha sido modificado de alguna manera.

Utilizando el comando grep, muestre por pantalla las líneas relacionadas con la palabra usb1 que genera la herramienta dmesg.

```
sansforensics@siftworkstation: ~  
$ dmesg | grep "usb1"  
[ 3.307331] usb usb1: New USB device found, idVendor=1d6b, idProduct=0001, bcdDevice= 4.18  
[ 3.307334] usb usb1: New USB device strings: Mfr=3, Product=2, SerialNumber=1  
[ 3.307349] usb usb1: Product: OHCI PCI host controller  
[ 3.307350] usb usb1: Manufacturer: Linux 4.18.0-15-generic ohci_hcd  
[ 3.307352] usb usb1: SerialNumber: 0000:00:06.0
```

Indica que el sistema ha detectado un nuevo dispositivo USB con el identificador de vendedor (idVendor=1d6b) y el identificador de producto (idProduct=0001).

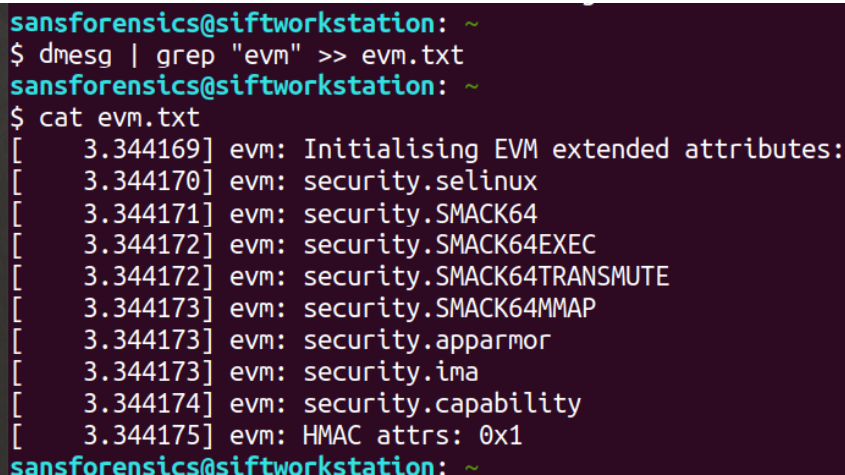
El dispositivo tiene cadenas de identificación para:

- Fabricante (Mfr=3)
- Producto (Product=2)
- Número de serie (SerialNumber=1)

Indica que el dispositivo USB detectado es un controlador host OHCI PCI, lo que sugiere que es un controlador de hardware que maneja conexiones USB.

Muestra que el sistema operativo Linux con el kernel 4.18.0-15-generic está gestionando este dispositivo a través del controlador ohci_hcd (Open Host Controller Interface para USB 1.1). Además, SerialNumber: 0000:00:06.0, como identificador único para el dispositivo

Genere un fichero llamado evm.txt con las líneas que tengan que ver con evm que devuelve dmesg.



```
sansforensics@siftworkstation: ~  
$ dmesg | grep "evm" >> evm.txt  
sansforensics@siftworkstation: ~  
$ cat evm.txt  
[ 3.344169] evm: Initialising EVM extended attributes:  
[ 3.344170] evm: security.selinux  
[ 3.344171] evm: security.SMACK64  
[ 3.344172] evm: security.SMACK64EXEC  
[ 3.344172] evm: security.SMACK64TRANSMUTE  
[ 3.344173] evm: security.SMACK64MMAP  
[ 3.344173] evm: security.apparmor  
[ 3.344173] evm: security.ima  
[ 3.344174] evm: security.capability  
[ 3.344175] evm: HMAC attrs: 0x1  
sansforensics@siftworkstation: ~
```

- **dmesg** → Comando que muestra los mensajes del buffer del kernel, es decir, registros del sistema desde el arranque.
- **grep "evm"** → Filtra solo las líneas que contienen la palabra "evm".
- **>> evm.txt** → Redirige la salida del comando a un archivo llamado evm.txt, sin sobrescribir su contenido si ya existe.

EVM es un módulo de seguridad en Linux que protege la integridad de los atributos de seguridad de los archivos.

[3.344169] evm: *Initialising EVM extended attributes*: El kernel está inicializando EVM y sus atributos extendidos.

[3.344170] evm: security.selinux: (Security-Enhanced Linux): módulo de seguridad que aplica control de acceso obligatorio.

[3.34417X] evm: security.SMACK64, EXEC, TRANSMUTE, MMAP: sistemas de control de acceso obligatorio. Se listan diferentes atributos de seguridad relacionados con SMACK.

[3.344173] evm: security.apparmor: sistema de control de acceso obligatorio utilizado en Linux.

[3.344174] evm: security.ima (Integrity Measurement Architecture): sistema para medir y verificar la integridad de los archivos en Linux.

[3.344174] evm: security.capability: control de los privilegios específicos que pueden tener los procesos en Linux.

[3.344175] evm: HMAC attrs: 0x1 (Hashed Message Authentication Code): proteger la integridad de los atributos de seguridad.

Esto es útil en análisis forense para verificar la integridad del sistema y detectar posibles manipulaciones en los atributos de seguridad de los archivos

Genere su huella digital en un fichero llamado huella_evm.txt

```
sansforensics@siftworkstation: ~  
$ sha256sum evm.txt > huella_evm.txt  
sansforensics@siftworkstation: ~  
$ cat huella_evm.txt  
8dae4ba5385241b791e0696b820bc075d44adfdebe4a077ccdf247dbdf5ed6a8  evm.txt  
sansforensics@siftworkstation: ~
```

Compruebe el funcionamiento de la orden wc contando el número de caracteres del fichero anterior y explique sus parámetros.

```
sansforensics@siftworkstation: ~  
$ wc -m huella_evm.txt  
74 huella_evm.txt  
sansforensics@siftworkstation: ~
```

wc -l: Devuelve el **número de líneas** de un fichero.

wc -w: Devuelve el **número de palabras** de un fichero (separa palabras por espacios en blanco, saltos de línea, tabulaciones, etc.).

wc -c: Devuelve el **número de bytes** de un fichero.

wc -m: Devuelve el **número de caracteres** (teniendo en cuenta la codificación).

wc -L: Devuelve la **longitud de la línea más larga** (en caracteres).

3. Actividades opcionales

Bloqueos de rangos de IP de Cloudflare por parte de Movistar ¿Cómo obtengo evidencias de que mi servicio esta caído/offline para poder tomar después medidas legales en contra de Movistar?

Una de las primeras evidencias que tomaría serían capturas de pantalla que incluiría en un archivo PDF del ping para saber el tiempo de respuesta del servidor, de traceroute para rastrear la ruta que sigue un paquete de datos a través de la red, identificando los nodos intermedios hasta su destino y de nslookup, para consultar servidores DNS y obtener la dirección IP asociada a un dominio y otros registros DNS relevantes.

Después con Wireshark recolectaría el tráfico de la red con archivos .pcap con su respectivo hash y guardaría los logs del servidor y logs de Cloudflare.

En todo momento hay que registrar la cadena de custodia de las evidencias y poner quién obtuvo la evidencia, la fecha y hora de adquisición y cómo se almacenó y protegió. Es importante que todas las evidencias que se recolecten se relacionen con timestamps para así poder demostrarlo todo de cara a un juicio.

Otras evidencias que se podrían recoger serían los intentos de acceso a la web desde red de Movistar en donde a parte de las capturas de pantalla que comentamos antes, habría que incluir los mensajes de error que nos diese el servidor de Movistar, y probar con otros operadores como Vodafone u Orange para poder comparar los resultados y ver si la conexión es exitosa. En ese caso habría que hacer capturas de la web cargada correctamente.

¿Cómo obtengo evidencias de que estas 3 aplicaciones tienen o no conexiones a la red en donde pueda estar filtrándose nuestro dni?

1. [SaferLayer](#)

2. [SafeID](#)

3. [Repositorio de Github \(PDF\)](#)

Para ver si esas aplicaciones establecen conexiones a internet y transfieren el documento a algún servidor lo primero sería verlo con las herramientas de desarrollador del navegador web pero no es algo muy profesional por lo que lo que haríamos nosotros sería aislar el entorno de prueba con una máquina virtual con Windows. En este punto habría que cerrar otras aplicaciones que generen tráfico como navegadores o aplicaciones de mensajería para reducir el ruido en la captura de red.

Cuando tengamos el entorno preparado habría ue realizar la acción de ponerle la marca de agua al dni y para recoger las primeras evidencias lo primero sería abrir Wireshark o TCPdump para capturar el tráfico de la red, tanto el entrante como el saliente.

También usaría herramientas de windows como sysmon para relacionar conexiones de red con procesos específicos, TCPview o Process Explorer para ver en tiempo real qué programas están estableciendo conexiones.

Otra cosa que podría hacerse que hemos visto posible es configurar un firewall de aplicaciones como GlassWire en Windows para recibir avisos cuando un programa intente comunicarse con la red. Si la aplicación utiliza HTTP/HTTPS y no tiene mecanismos muy fuertes de ofuscación, se podrían ver las solicitudes que hace, por lo que esto puede evidenciar si el DNI se está subiendo a un servidor remoto.

Todas estas evidencias, logs y capturas de wireshark deben tener siempre las marcas de tiempo con fechas y horas exactas, además del hash correspondiente a cada una de ellas (para demostrar integridad) para poder documentarlas correctamente en el documento de la cadena de custodia.

Por último, para hacer un análisis más profundo se podría revisar el binario con herramientas como strings, IDA o Ghidra que hemos visto en la asignatura de análisis de malware, pero esto va iría más allá del monitoreo de red.