RETO ANALISIS FORENSE
FORO.ELHACKER.NET

1. Introducción.
El reto de análisis forense objeto de este estudio ha sido proporcionado por el equipo del foro.elhacker.net.
La misión es realizar un estudio de una imagen de un pendrive para contestar a las preguntas que se detallan en el siguiente apartado y entregar un informe técnico exhaustivo con todos los pasos seguidos durante la investigación.

2. Información de referencia.
Ésta es la información proporcionada para la resolución del caso:

> La Brigada Especial de Delitos Informáticos ha sido requerida por la Policía.
>
> Ésta, llevaba 2 meses detrás de una red de narcotraficantes, consiguiendo averiguar que en unos pocos días va a tener lugar uno de los intercambios de cocaína más importantes de los últimos tiempos.
>
> En el momento oportuno atraparon a uno de los integrantes de la banda residente en el país, el cual llevaba en su poder un pendrive con información que se piensa puede ser muy valiosa para el desmantelamiento de ambas bandas y del intercambio.
>
> En este caso te ha tocado a ti investigar el caso, una gran responsabilidad, pero a la vez una gran oportunidad, serás capaz de estar a la altura??
>
> Los "expertos" de la policía han hecho sus propios hallazgos antes de entregarte el pendrive:
> - Sistema de Archivos: ext2
> - Información de utilidad en el penDrive: fecha, hora y lugar del intercambio (se lo consiguieron sonsacar al detenido, así como el conocimiento de que todo lo que se pueda necesitar en un momento dado, está igualmente en el pendrive)
>
> Por desgracia parece ser que los delincuentes fueron precavidos, y tomaron ciertas medidas para que esta información no sea evidente a primera vista...
>
> Tu misión será hacer un análisis exhaustivo de la situación para conseguir sacar todos los datos posibles: Fecha y hora de la entrega, nombre del jefe de la banda, lugar del intercambio...

3. Preguntas a contestar.
El objetivo es obtener la siguiente información:
1.Fecha y hora de la entrega.
2.Nombre del jefe de la banda.
3.Lugar del intercambio.

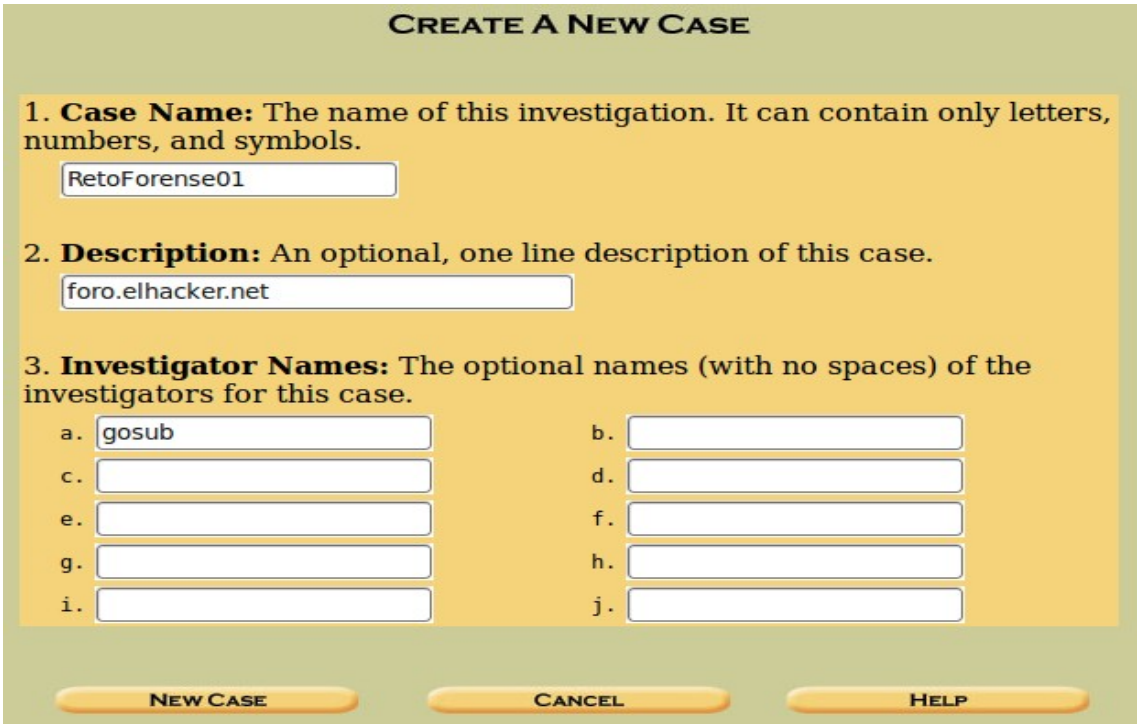
4. Análisis.
El primer paso consiste en descargar la imagen proporcionada junto con la información de la suma de control md5: sdb1.dd y sdb1.md5.
A continuación verificamos la integridad del archivo:

```
#md5sum -c sdb1.md5
sdb1.dd: La suma coincide
```

Una vez confirmada su integridad abrimos un nuevo caso en Autopsy, el cual previamente hemos descargado y configurado en la maquina donde se realiza el análisis.

Ventana de inicio.

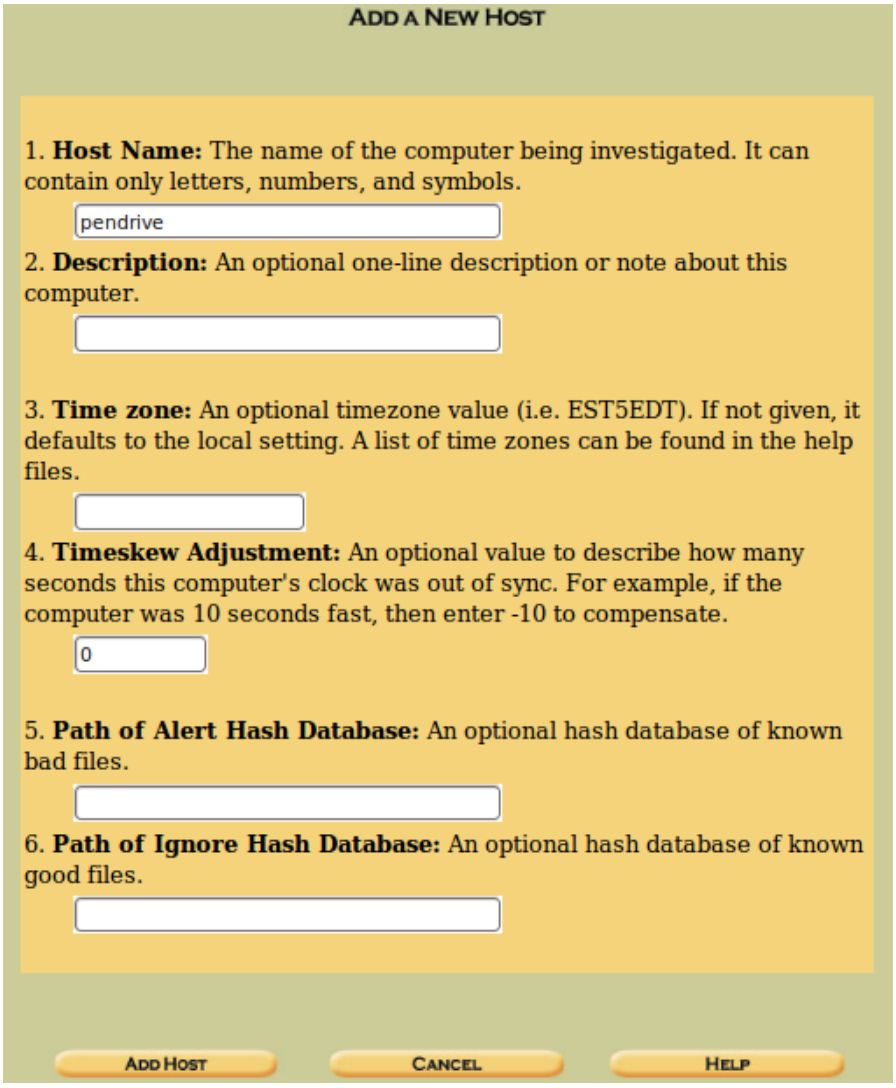
En la ventana de inicio seleccionamos la opción “New Case” e introducimos los datos correspondientes:



Creación de un caso nuevo

Una vez configurados los valores el programa crea los correspondientes directorios donde se almacenará toda la información del caso:

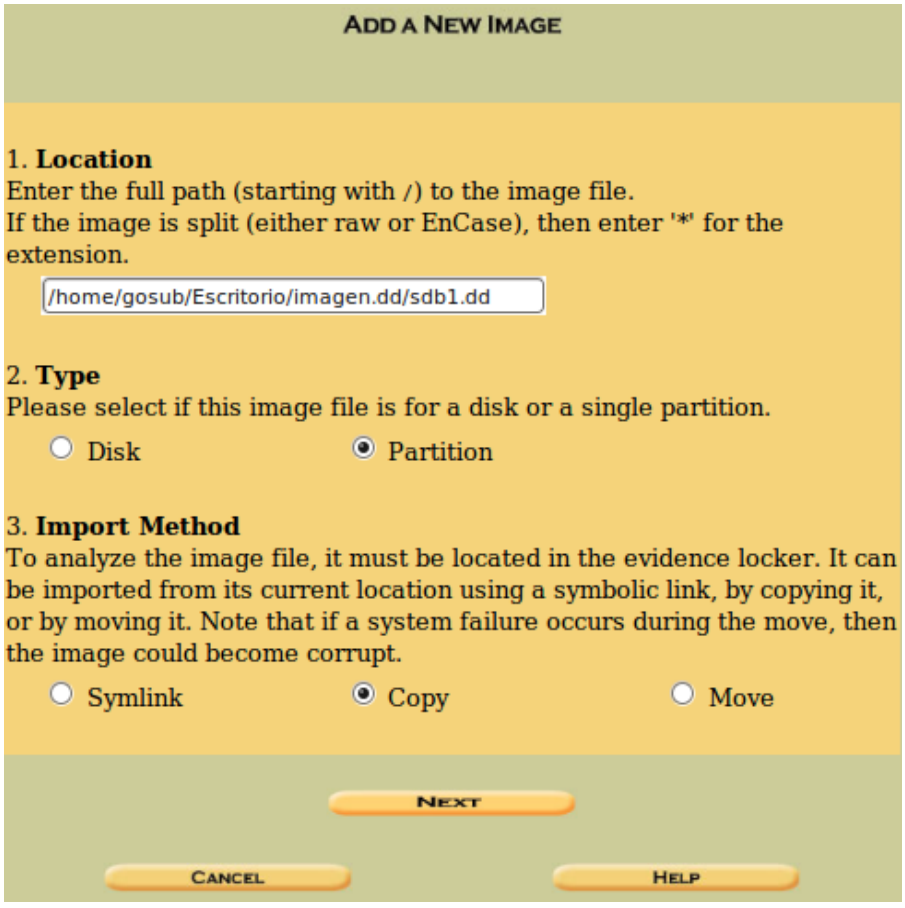
Seleccionamos el botón “Add Host” y nos conduce a la siguiente pantalla de configuración donde daremos los datos que nos solicita.



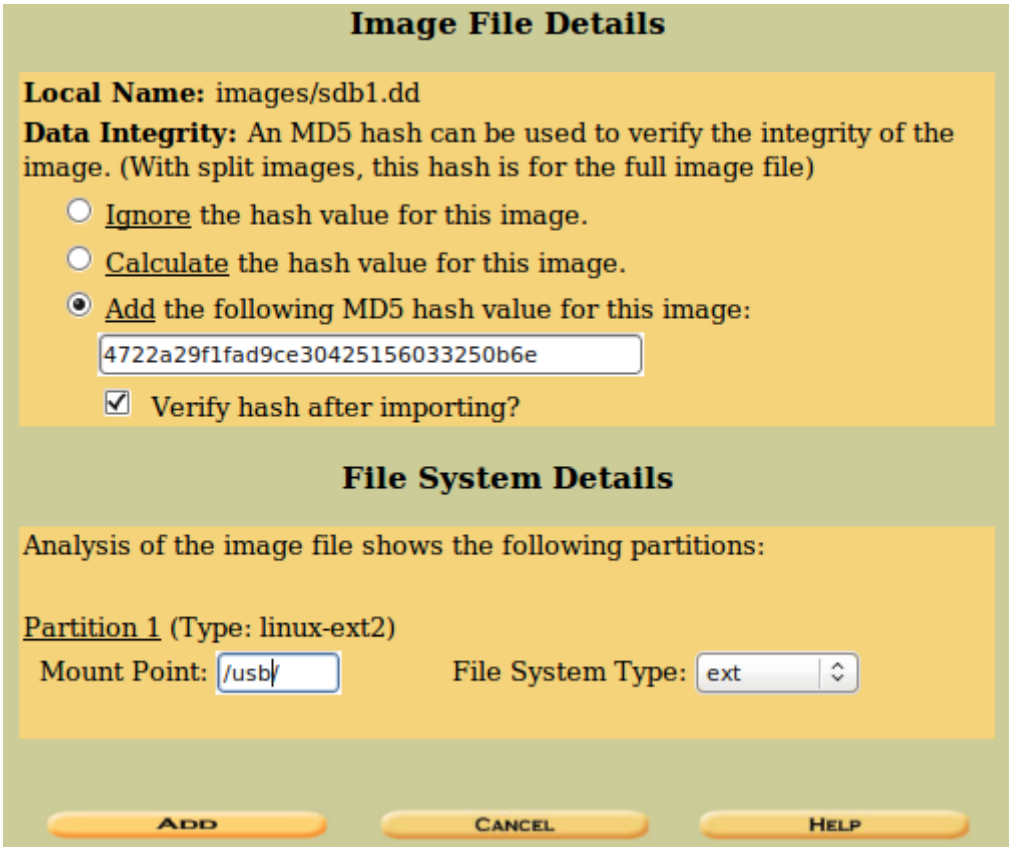
Como no se trata de la imagen de un disco duro de un ordenador sino de un pendrive introducimos esto como nombre. La zona horaria sabemos que es la misma y no tenemos información sobre si la máquina donde se creó el pendrive tenía algun desajuste en la hora.
No se nos ha proporcionado ninguna base de datos de alerta o ignorar por lo que también lo dejamos en blanco y seleccionamos “Add Host”.

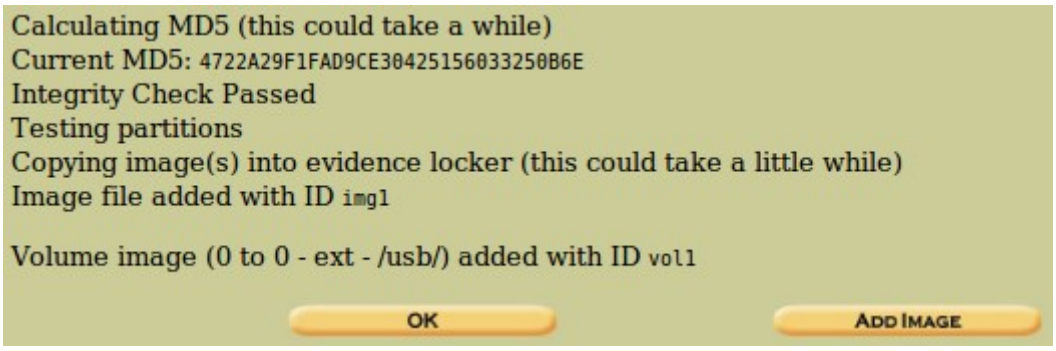
Ahora añadimos al caso la imagen del pendrive pulsando en “Add Image” y a continuación “Add Image File”:

El programa nos pide a continuación la imagen a incorporar al caso y el método de importación que preferimos:

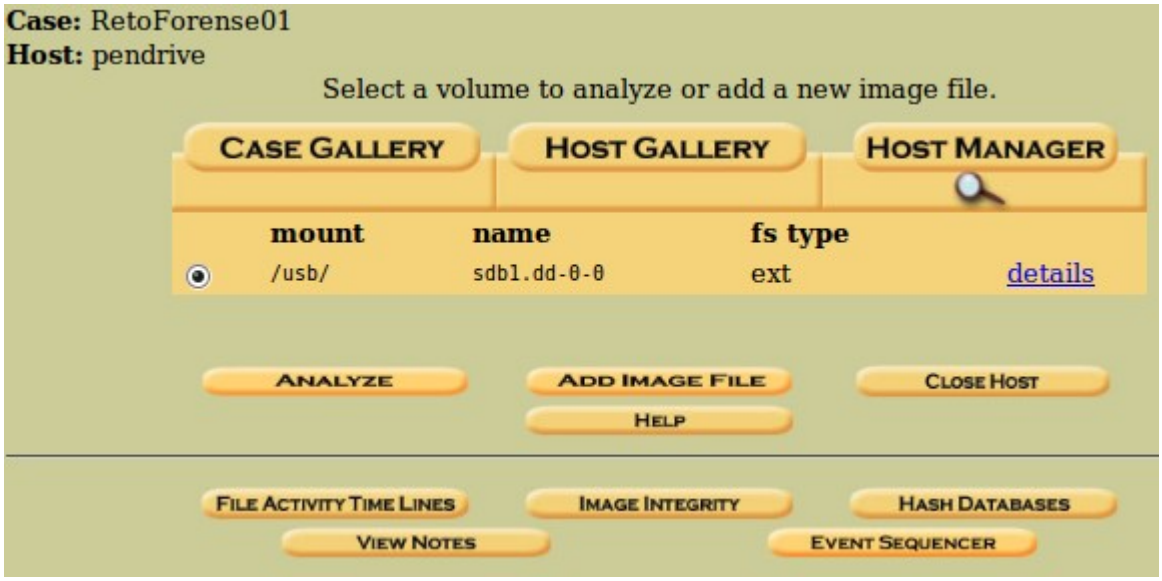


Seleccionamos “Next” y nos solicita los detalles de la imagen. Como disponemos del hash md5 lo proporcionamos y le decimos que lo verifique después de importar la imagen:



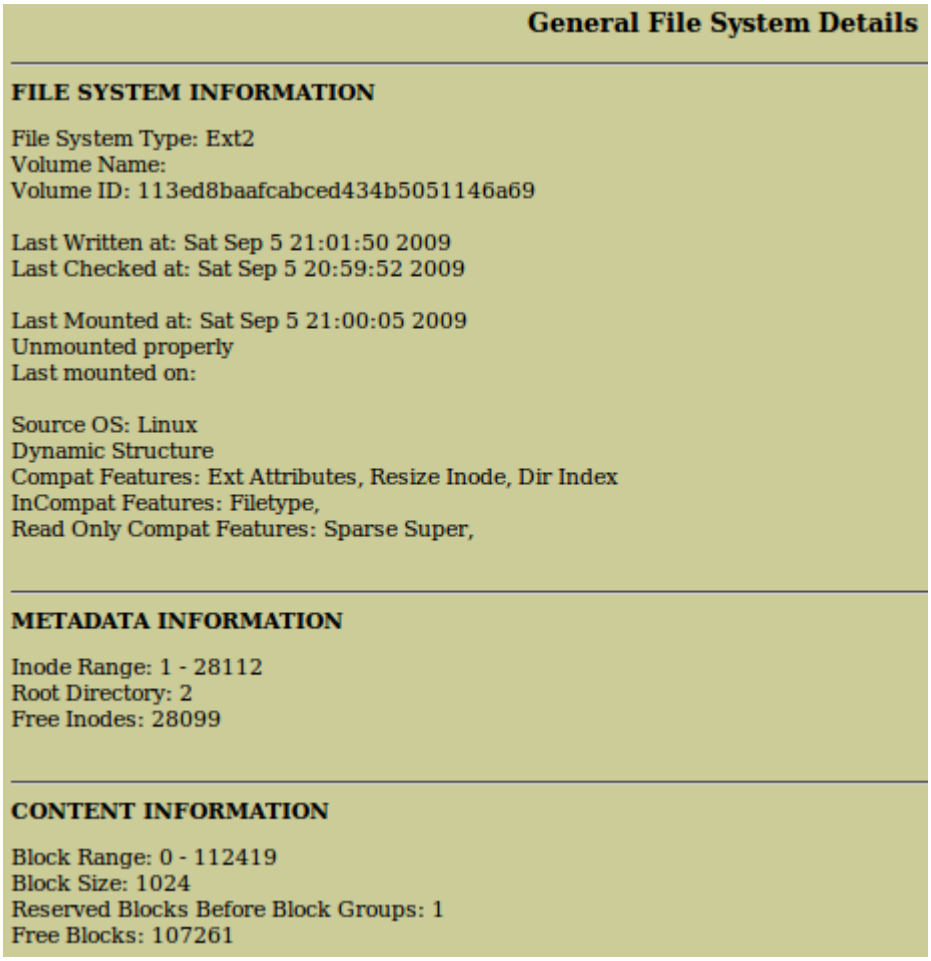


Después de verificar que la imagen ha sido importada correctamente ya podemos pulsar “OK” para ir a la ventana principal de gestión de casos “Host Manager” donde podemos comenzar con el análisis del pendrive.



Para comenzar el análisis pulsamos “Analyze” y aparece la página principal de análisis.

Pulsamos primero en “Image Details” donde recabamos información sobre el sistema de archivos. Tomo nota del tamaño de bloque, 1024 bytes, ultimas fechas de acceso, tamaño total, que coincide con el tamaño de la imagen descargada y el espacio libre.



Seleccionamos ahora “File Analysis” del menú.

Del	Type dir / in	Name	Modified	Accessed	Changed	Size	UID	GID	Meta
	d / d	../	2009.09.05 21:01:41 (CEST)	2009.09.05 21:01:40 (CEST)	2009.09.05 21:01:41 (CEST)	1024	0	0	2
	d / d	./	2009.09.05 21:01:41 (CEST)	2009.09.05 21:01:40 (CEST)	2009.09.05 21:01:41 (CEST)	1024	0	0	2
	r / r	jlo.jpg	2009.09.05 21:00:35 (CEST)	2009.09.05 21:01:23 (CEST)	2009.09.05 21:00:35 (CEST)	43769	0	0	12
	d / d	lost+found/	2009.09.05 20:59:52 (CEST)	2009.09.05 20:59:52 (CEST)	2009.09.05 20:59:52 (CEST)	12288	0	0	11
✔	r / -	mail	0000.00.00 00:00:00 (GMT)	0000.00.00 00:00:00 (GMT)	0000.00.00 00:00:00 (GMT)	0	0	0	0
	r / r	script.sh	2009.09.05 21:01:05 (CEST)	2009.09.05 21:01:05 (CEST)	2009.09.05 21:01:05 (CEST)	10105	0	0	14

Nos presenta el listado de los archivos y directorios encontrados en el volumen. Inmediatamente se observa un archivo llamado “mail” que no se puede recuperar, un archivo llamado “jlo.jpg”, otro llamado “script.sh” y un directorio “lost+found”. Dentro de este directorio no encontramos ningún archivo.
Como primera opción pulsamos sobre el archivo “jlo.jpg” y vemos que no se reconoce como un archivo de imagen, presentando una cabecera de archivo solo con ceros.

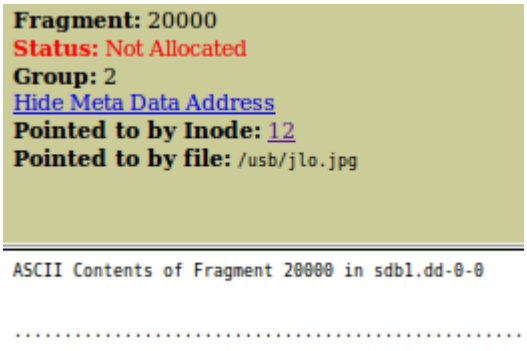
ASCII (display - report) * Hex (display - report) * ASCII Strings (display - report) * Export * Add Note
File Type: data

```
Hex Contents Of File: /usb/jlo.jpg

00000000:  0000 0000 0000 0000 0000 0000 0000 0000    ................
00000010:  0000 0000 0000 0000 0000 0000 0000 0000    ................
00000020:  0000 0000 0000 0000 0000 0000 0000 0000    ................
00000030:  0000 0000 0000 0000 0000 0000 0000 0000    ................
00000040:  0000 0000 0000 0000 0000 0000 0000 0000    ................
00000050:  0000 0000 0000 0000 0000 0000 0000 0000    ................
00000060:  0000 0000 0000 0000 0000 0000 0000 0000    ................
```

Vamos a hechar un vistazo a los meta datos del archivo pulsando sobre el valor “12” que aparece en la columna “Meta” situada a su derecha.

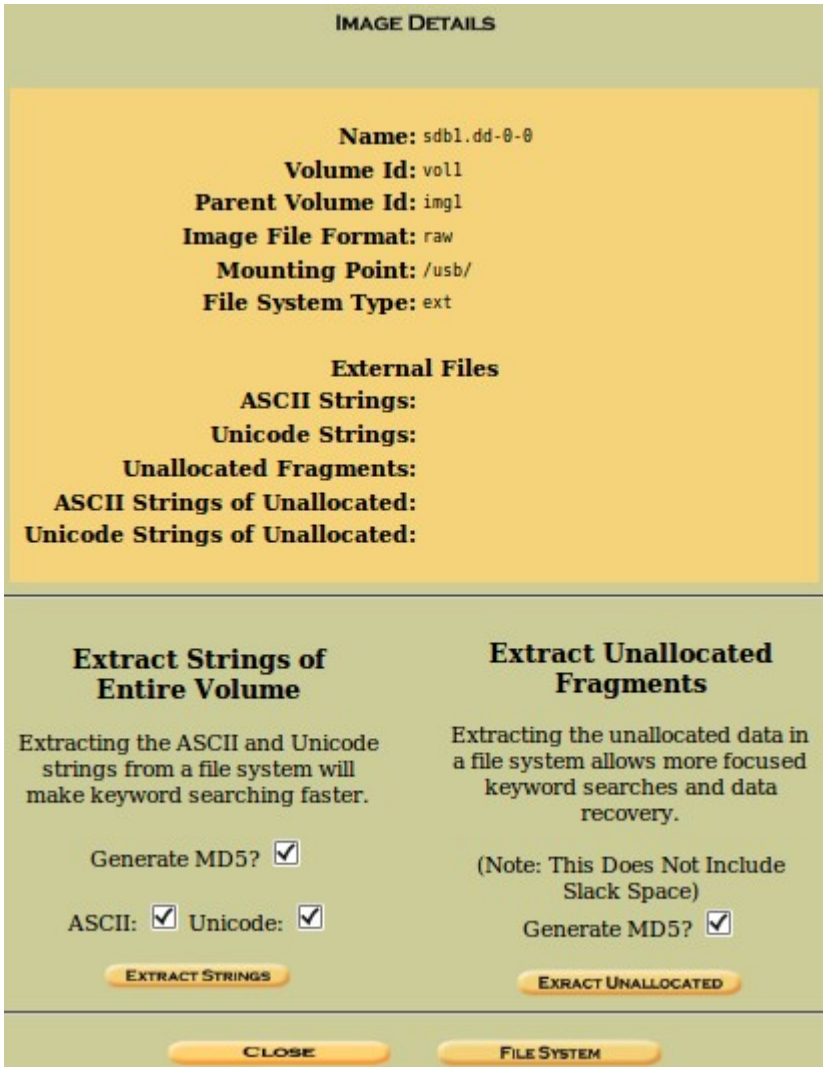
Siempre que entremos en esta vista de un archivo realizaremos la comprobación de que efectivamente el tamaño del archivo coincide con el número total de bloques asignados puesto que ya sabemos cual es el tamaño de bloque del volumen.
A la vista de los bloques del archivo llama la atención el bloque 20000, que parece fuera de sitio, y el bloque 7693, que rompe la secuencia de la asignación de bloques. Pulsando sobre el bloque 20000 obtenemos la información de que se trata de un espacio no asignado y que está vacío.



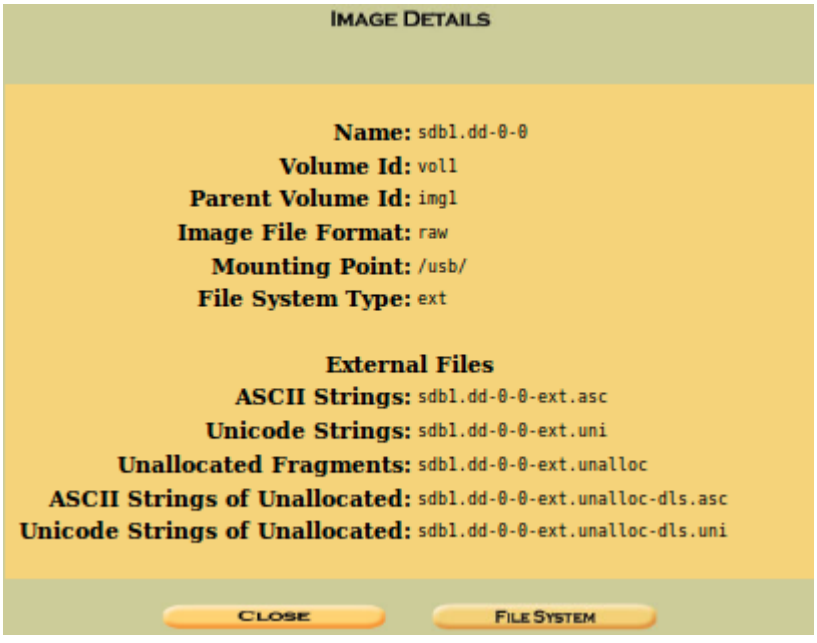
Si realmente se trata de una imagen jpg deberá existir una cabecera que comience tipicamente, aunque hay otras cabeceras posibles, por

```
FF D8 FF E0 xx xx 4A 46        ÿØÿà..JFIF
```

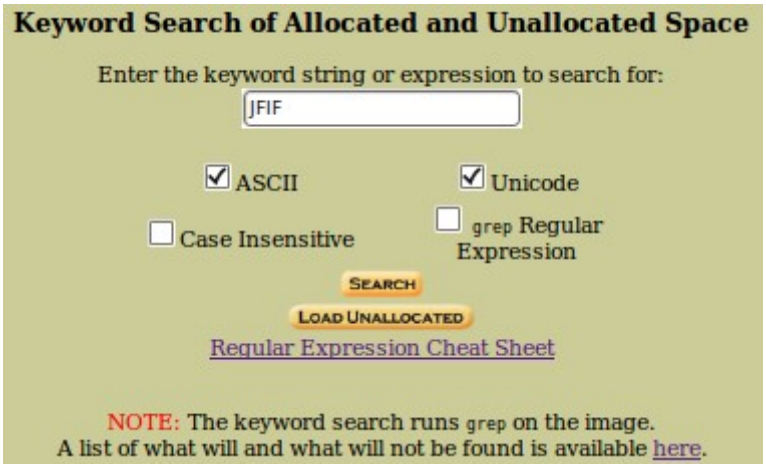
por lo que vamos a hacer una búsqueda. Pero antes vamos a extraer las cadenas ASCII y Unicode. Para ello pulsamos en el menú superior sobre el botón “Close” y volvemos a la pantalla “Host Manager”. Seleccionamos el enlace “details” a la derecha del volumen.



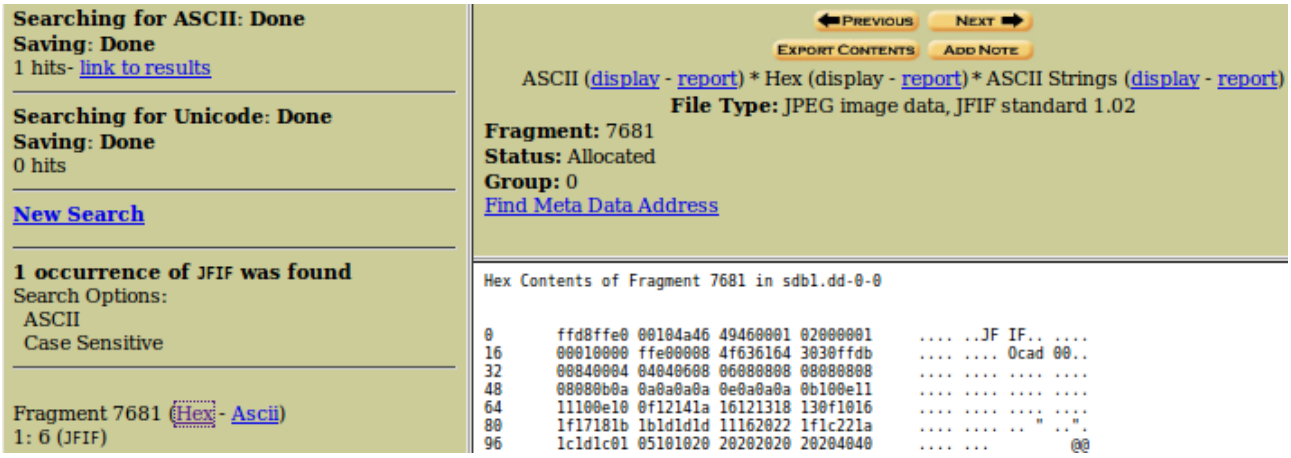
Lo que vamos a hacer es extraer las cadenas de texto ASCII y Unicode para acelerar las búsquedas que tengamos que realizar en el volumen. Primero lo haremos en el sistema de archivos del volumen y a continuación de los fragmentos no asignados pulsando primero sobre “Extract Strings” y a continuación sobre “Extract Unallocated”.



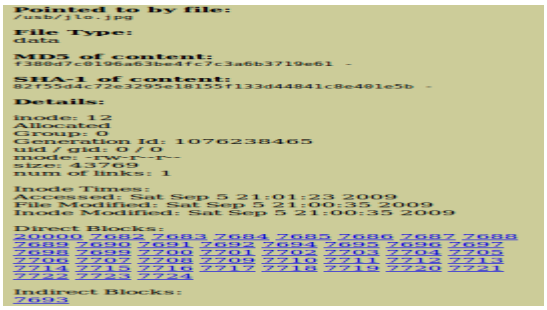
Ya podemos regresar a la página principal de análisis y pulsamos sobre “Keyword Search”.



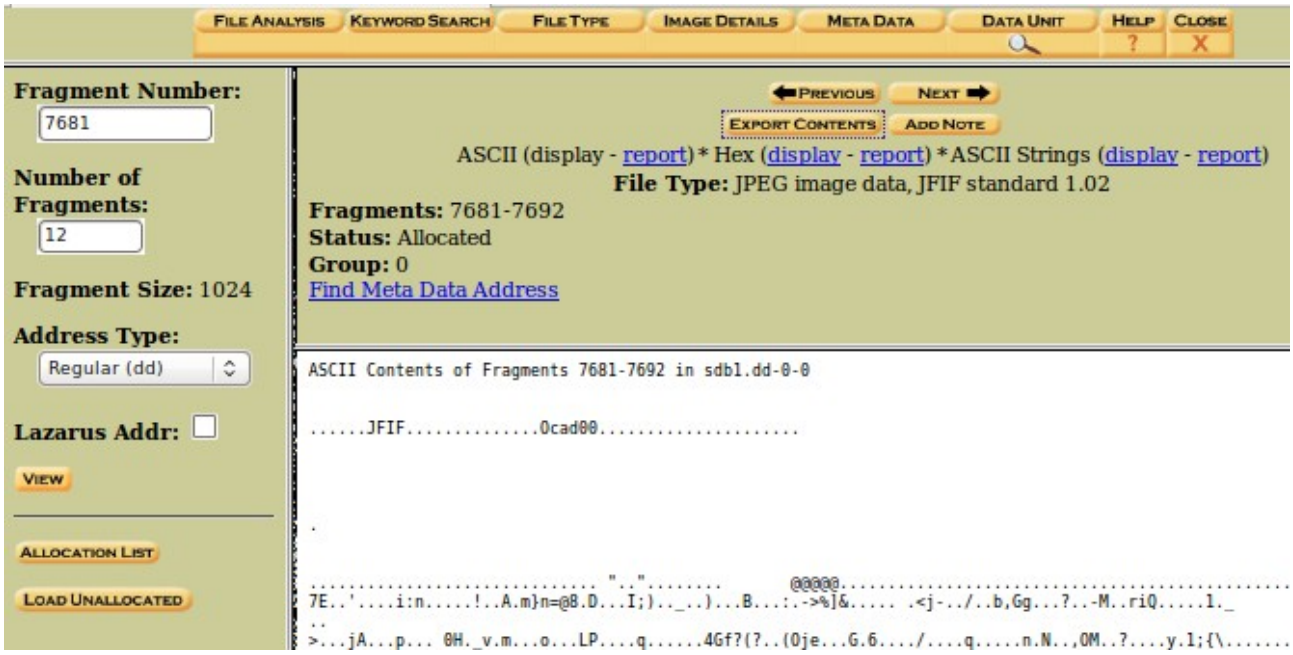
Introducimos JFIF en el cuadro de búsqueda y pulsamos sobre “Search”.



Encontramos una única ocurrencia de JFIF que está situada en el bloque 7681, y si recordamos la estructura de bloques del archivo:



Vemos que es justamente el bloque que intuitivamente le faltaba al archivo. Bien, lo que haremos será extraer los bloques y ver si tienen sentido. Para acceder al contenido de los 43 bloques seleccionamos en el menú superior el botón “Data Unit”. En la casilla “Fragment Number” introducimos el valor 7681 y en la de “Number of Fragments” el valor 12 y pulsamos “View”. Lo que pretendemos es extraer los 43 bloques que ocupa el archivo pero evitando el bloque 7693, que no forma parte del archivo, por lo que extraeremos desde el 7681 hasta el 7692 y a continuación desde el 7694 hasta el 7724.



Pulsamos sobre el botón “Export Contents” y guardamos el archivo como <<vol1-Fragment7681.raw>>. Ha continuación introducimos los valores de 7694 y 31 y mediante “Export Content” salvamos el archivo <<vol1-Fragment7694.2.raw>>.
Desde una terminal ejecutamos, en la carpeta donde se encuentren los archivos anteriormente salvados, la orden

```
$cat vol1-Fragment* > jlo-rescatada.jpg
```

Y obtenemos la siguiente imagen:

Normalmente en este tipo de casos se emplean las imagenes como medio para ocultar información mediante métodos esteganográficos. Para detectar la forma más elemental de ocultación pulsamos, en las ventanas donde seleccionamos los bloques anteriormente, sobre el botón “display” situado a la derecha en “ASCII Strings (display – report)” y hechamos un vistazo a las cadenas.

Antes de usar otros metodos de detección esteganográficos como “stegdetect” vamos a hechar un vistazo al otro archivo existente en la imagen.

Del	Type dir / in	Name	Modified	Accessed	Changed	Size	UID	GID	Meta
	d / d	../	2009.09.05 21:01:41 (CEST)	2009.09.05 21:01:40 (CEST)	2009.09.05 21:01:41 (CEST)	1024	0	0	2
	d / d	./	2009.09.05 21:01:41 (CEST)	2009.09.05 21:01:40 (CEST)	2009.09.05 21:01:41 (CEST)	1024	0	0	2
	r / r	jlo.jpg	2009.09.05 21:00:35 (CEST)	2009.09.05 21:01:23 (CEST)	2009.09.05 21:00:35 (CEST)	43769	0	0	12
	d / d	lost+found/	2009.09.05 20:59:52 (CEST)	2009.09.05 20:59:52 (CEST)	2009.09.05 20:59:52 (CEST)	12288	0	0	11
✔	r / -	mail	0000.00.00 00:00:00 (GMT)	0000.00.00 00:00:00 (GMT)	0000.00.00 00:00:00 (GMT)	0	0	0	0
	r / r	script.sh	2009.09.05 21:01:05 (CEST)	2009.09.05 21:01:05 (CEST)	2009.09.05 21:01:05 (CEST)	10105	0	0	14

ASCII (display - report) * Hex (display - report) * ASCII Strings (display - report) * Export * Add Note
File Type: OpenDocument Text (gzip compressed data, was "place.odt", from Unix, last modified: Fri Sep 4 15:37:14 2009)

Contents Of File: /usb/script.sh

De momento parece que se trata de un archivo gzip con un documento en su interior denominado place.odt del tipo OpenDocument Text. Hechamos un vistazo a los metadatos correspondientes pulsando sobre el enlace del inodo a la derecha del archivo.

Pointed to by file:
/usb/script.sh

File Type:
OpenDocument Text (gzip compressed data, was "place.odt", from Unix, last modified: Fri Sep 4 15:37:14 2009)

MD5 of content:
1eb4046ac4bfd52821c5395967fd2667 -

SHA-1 of content:
f7e4de06937412cff6617390593273d9fecb4860 -

Details:

inode: 14
Allocated
Group: 0
Generation Id: 1076238467
uid / gid: 0 / 0
mode: -rw-r--r--
size: 10105
num of links: 1

Inode Times:
Accessed: Sat Sep 5 21:01:05 2009
File Modified: Sat Sep 5 21:01:05 2009
Inode Modified: Sat Sep 5 21:01:05 2009

Direct Blocks:
527 528 529 530 531 532 533 534
535 0

De nuevo observamos una estructura extraña en los bloques ocupados por el archivo porque el décimo, en lugar de ser el 536 resulta ser el 0. Pulsando sobre él hechamos un vistazo a su contenido resultando estar vacío. Si acudimos a la ventana de “Data Unit” y buscamos el bloque 536 encontramos contenido.
Mediante el procedimiento anteriormente descrito extraemos el archivo de las dos formas: acabando los últimos 1024 bytes con el contenido del bloque 0 y por otro lado con el contenido del bloque 536. Al tratar de abrir el primer archivo obtenemos un error

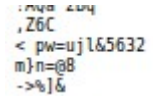
```
$ gzip -d vol1-Fragment527-0.gz
gzip: extract.gz: unexpected end of file
```

Mientras que con el segundo:

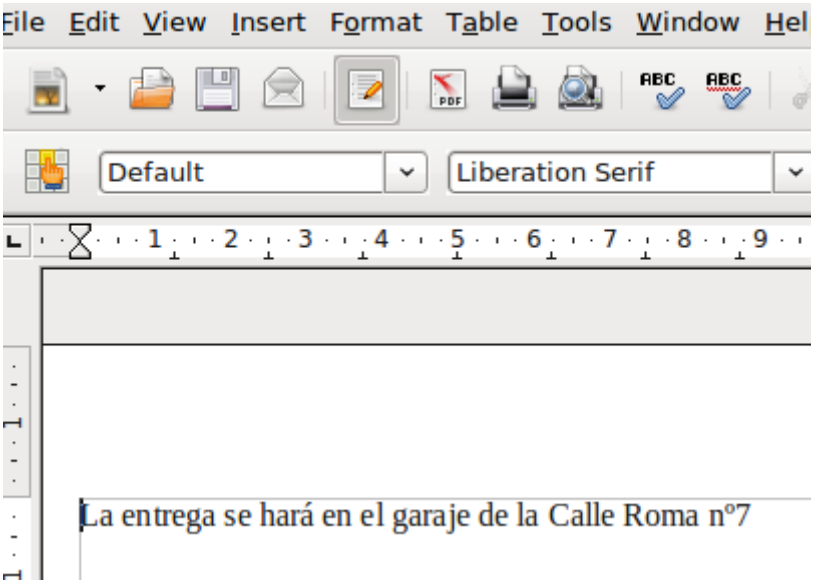
```
$ gzip -d vol1-Fragment527-536.gz
```

Obtenemos otro archivo llamado <<vol1-Fragment527-536>>.
Lo renombramos a <<place.odt>> y lo tratamos de abrir obteniendo

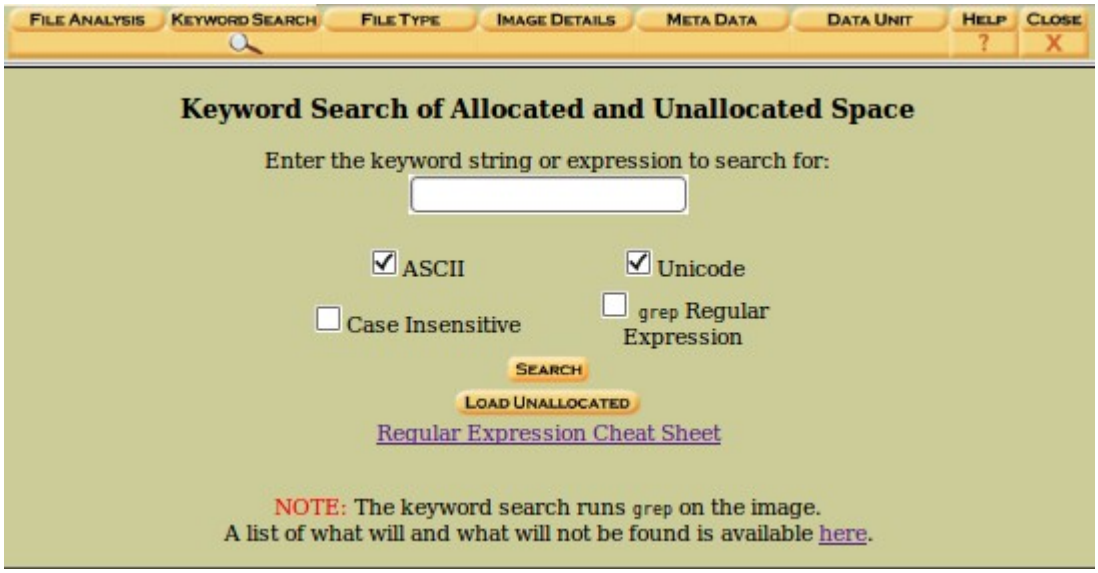
Es hora de volver a examinar las cadenas de texto de la imagen extraida anteriormente y al hacerlo llama la atención una linea



que parece significativa: pw=ujl&5632
Y en efecto es la password buscada obteniendo el siguiente documento:



De momento parece que los dos archivos legibles nos han ofrecido toda la información que contenían. Es hora de hechar un vistazo al resto del espacio de la unidad.
Volvemos a la página principal de análisis y entramos en “Keyword Search”.



Seleccionamos la búsqueda predefinida “Date” y obtenemos un resultado que parece corresponder con un mensaje ya borrado.

5. Respuestas.

Hora de la entrega: 02:00

Nombre del jefe de la banda: Osvaldo.

Lugar del intercambio: En el garaje de la calle Roma nº 7.