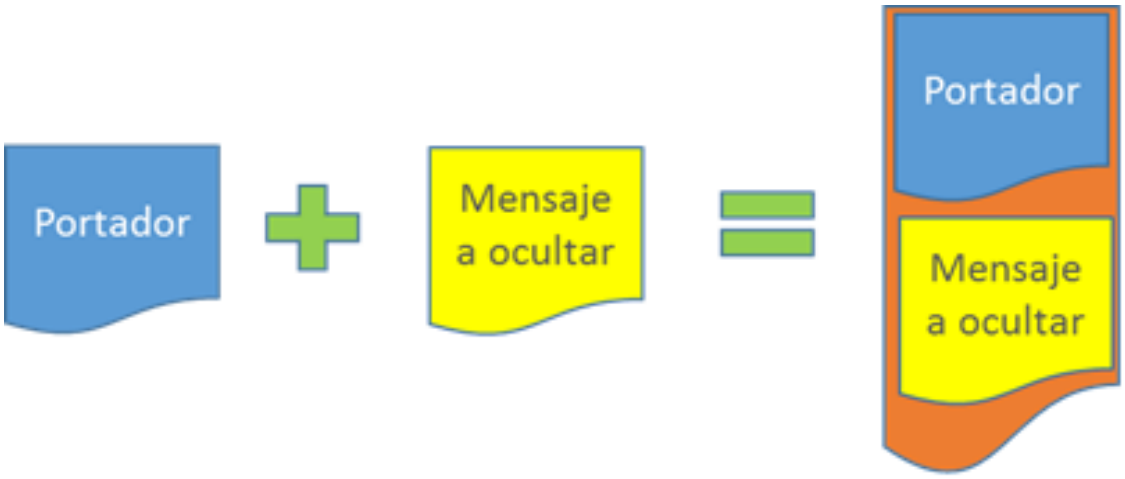
ESTEGANOGRAFÍA FÁCIL. CASO PRÁCTICO

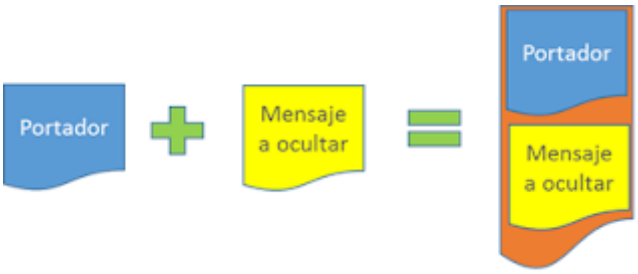


La esteganografía consiste en la aplicación de técnicas para ocultar un mensaje dentro de otro, conocido como portador. El término proviene del griego steganos (oculto) y graphos (escritura). En el fondo se trata de algo tan aparentemente sencillo como es esconder algo a la vista de todos.

En el mundo de la informática se puede esconder cualquier tipo de archivo siendo los portadores típicos documentos multimedia de audio, video o imagen.

LA OCULTACIÓN

Hoy vamos a ver un caso de ocultación esteganográfica muy sencilla usando el método de concatenación conocido como appendX. Este método consiste en algo tan sencillo como "añadir" al final del archivo portador el archivo a ocultar:



La forma de hacer esto es muy sencilla. Si estemos en un entorno Windows solo tenemos que ejecutar en el símbolo del sistema, en la carpeta donde tengamos guardados los documentos portador y a ocultar lo siguiente:

```
copy /b portador.jpg+fichero_a_ocultar.zip escondido.jpg
```

Tras la ejecución de este comando obtendremos el archivo escondido.jpg. Si lo visualizamos no veremos más que la imagen del portador.

Ejemplo:

Tengo el fichero White_Cat_and_a_mouse.jpg que voy a usar como portador y el 201_military_base_08.jpg que quiero ocultar. El comando a ejecutar sería:

```
copy /b White_Cat_and_a_mouse.jpg+201_military_base_08.jpg lindogatito.jpg
```

```
W:\>copy /b White_Cat_and_a_mouse.jpg+201_military_base_08.jpg lindogatito.jpg
White_Cat_and_a_mouse.jpg
201_military_base_08.jpg
        1 archivo(s) copiado(s).
```

Y si miro la carpeta con el explorador de Windows tendré lo siguiente:

Vemos que la imagen final del copiado lindogatito.jpg es indistinguible de la elegida como portadora.

Vamos a ocultar también un archivo zip en otra imagen:

```
W:\>copy /b 11900550834_ef635d5b5c_z.jpg+Texto.zip bonitogatito.jpg
11900550834_ef635d5b5c_z.jpg
Texto.zip
        1 archivo(s) copiado(s).
```

Siendo el resultado como anteriormente indistinguible de la imagen portadora:

Obviamente si hay una diferencia: el tamaño del archivo, pero si eliminamos del ordenador la imagen original no tendremos con que compararla y nos parecerá una imagen sin más.

En sistemas Linux/Unix es también muy sencillo. Se consigue el mismo resultado usando el comando "cat":

```
cat portador.jpg ficheroaesconder.zip > ficheroescondido.jpg
```

Hay otros métodos, pero este es indudablemente el mas sencillo de todos.

EL ANÁLISIS

Ahora imaginemos el siguiente escenario: estamos realizando un análisis forense a un disco duro del que previamente hemos hecho una copia bit a bit del mismo. No es necesario decir que tenemos que usar una de las copias de ese disco y no el original.

Analizando su contenido nos encontramos con varias imágenes que nos parecen sospechosas, ¡quien guarda en su HD fotos de gatos!!!...bueno…no es tan sospechoso…

¿CÓMO AVERIGUAMOS SI DETRÁS DE ESOS LINDOS GATITOS NUESTRO SOSPECHOSO ESTÁ ESCONDIENDO ALGO?

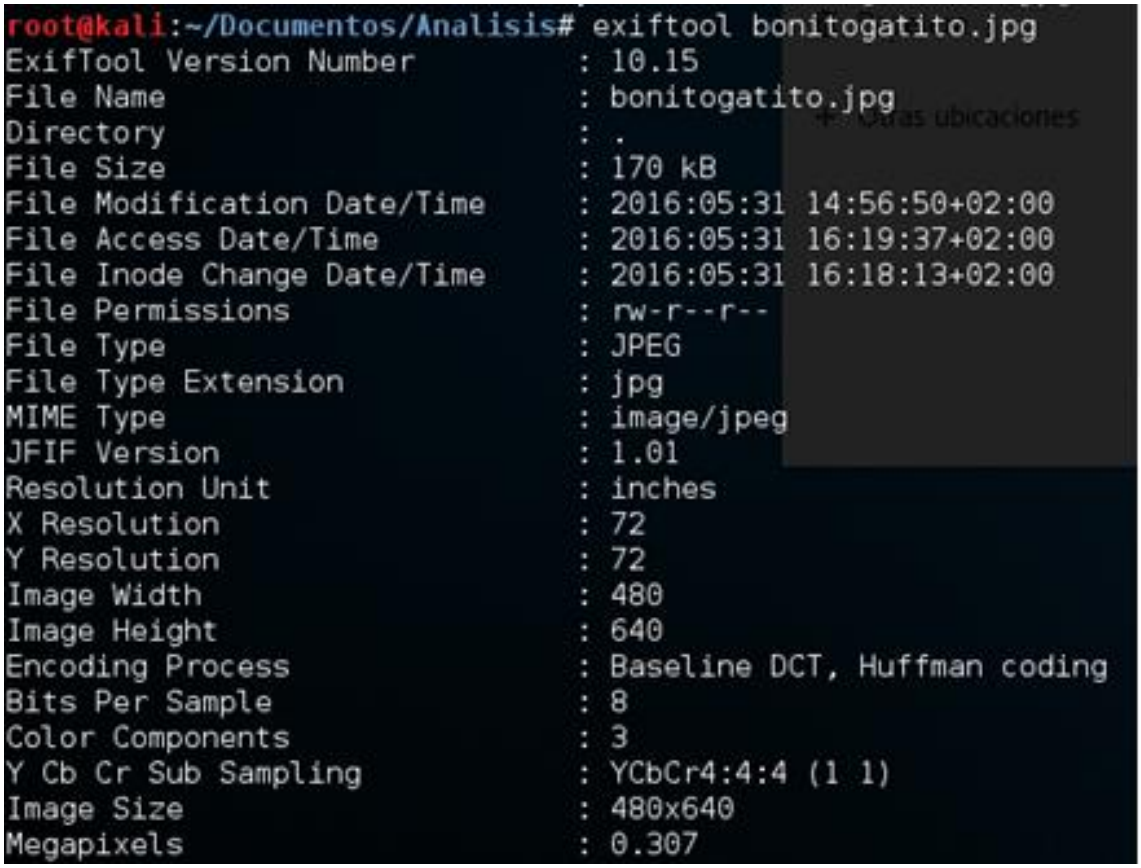
He decidido usar Kali Linux para hacer el análisis forense de las imágenes. Si las visualizo sin más con el visor de archivos veo lo mismo que veía en Windows:

Lo primero que intento, es obtener más información de las imágenes leyendo los metadatos. Para hacer esto uso la herramienta exiftools que previamente instalo en Kali con el comando:

```
apt-get install exiftool
```

Recordar que Kali recomienda usar por defecto el usuario root, por lo que no es necesario hacer sudo.

Buscando información en los metadatos de la primera imagen (bonitogatito.jpg) no veo nada raro:



El análisis de metadatos no da ninguna pista. Usamos otro camino. Instalamos la herramienta stegdetect, que nos permite averiguar si una imagen tiene información oculta y que método de ocultación se ha usado. Tener cuidado con esta herramienta, no detecta todos los posibles métodos de ocultación ya que se ha quedado un poco desfasada, por ejemplo, no detecta el método jphide usado por el programa de ocultación steghide.

Instalar esta herramienta, stegdetect, requiere de un par de pasos. Desde un terminal en Kali ejecutar:

```
wget http://archive.debian.org/debian/pool/main/s/stegdetect/stegdetect_0.6-3_amd64.deb
```

```
dpkg -i stegdetect_0.6-3_amd64.deb
```

Con esto tendremos instalada la herramienta que vamos a usar a continuación.

Desde el directorio donde tengamos las imágenes a analizar ejecutamos lo siguiente:

```
stegdetect *.jpg
```

Obtenemos el siguiente resultado:

```
root@kali:~/Documentos/Analisis# stegdetect *.jpg
bonitogatito.jpg : appended(3806)<[random][data][PK........Nc.H..]>
lindogatito.jpg : appended(3306)<[nonrandom][data][......Exif..II*.]>
Otro gato.jpg : negative
Otro gato mas.jpg : negative
```

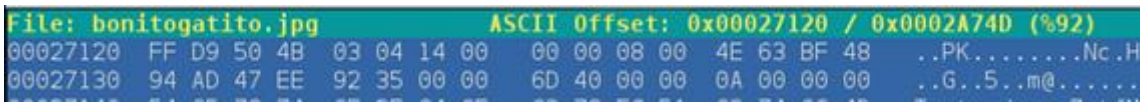
Esto ya si que nos da información: las dos primeras imágenes tienen datos añadidos al final. La primera aparentemente esconde un fichero con cabecera PK, lo que quiere decir que puede ser un .zip, y el segundo un Exif II que es una imagen.

LA EXTRACCIÓN DE LA INFORMACIÓN OCULTA

Pues ahora que sabemos que tenemos dos imágenes portadoras de archivos ocultos vamos a ver cómo podemos separar ambos.

El proceso es prácticamente el mismo en ambos casos. Lo primero es proceder a examinar con un editor/visor hexadecimal cada uno de los archivos para buscar las cabeceras y finales esperados para la extensión. En nuestro caso, tratándose de ficheros .jpg buscaremos en hexadecimal 0xFFD8 (SOI: Start Of Image) como cabecera y 0xFFD9 (EOI: End Of Image) como final. La información existente después de este final será el fichero oculto tras la imagen.

Usaremos como editor la herramienta hexeditor que trae Kali. Una vez abierto el fichero que queremos analizar usaremos el buscador (ctrl+W) para encontrar los bytes en hexadecimal correspondientes a EOI (FF D9). El resultado es el siguiente:



Aquí ya tenemos mucha información:

1. La dirección del último byte del archivo contenedor es la 0x27121. Recordar que la columna de la izquierda nos da la dirección de la posición del primer byte de la línea que precede en hexadecimal. En este caso el byte FF está en la posición “00027120”.
2. Derivado de este dato podemos saber que la longitud del archivo portador es en hexadecimal 0x27122 (recordar que el primer byte está en la dirección 0x00, no en la 0x01. Convirtiendo este valor a decimal, por ejemplo, con la calculadora de Windows o con en Linux con `echo $((16#27122))`, nos da 160034, que es la longitud en decimal del portador.
3. Todo indica que el archivo oculto en la imagen es un zip ya que la cabecera es 0x504B, o PK en ASCII.

Ahora solo nos queda extraer el fichero oculto tras el portador. Para ello vamos a usar el comando dd de Linux de la siguiente manera:

```
dd if=bonitogatito.jpg of=oculto.zip bs=1 skip=160034
```

Los parámetros usados son:

if: es el fichero de entrada, es decir, la imagen que sabemos que tiene un archivo oculto añadido

of: es el fichero de salida, al que le ponemos la extensión zip porque sabemos que ese es su contenido por la cabecera PK.

bs: el el tamaño del bloque para el siguiente argumento, en este caso 1 Byte

skip: le decimos que “esquive” 160034 bloques (es decir, bytes tal y como hemos indicado con el parámetro anterior) y que escriba en el fichero de salida a partir de esa posición.

El resultado es un fichero llamado oculto.zip, que si analizamos con unzip nos confirma que estamos en lo cierto en cuanto al contenido oculto:

```
root@kali:~/Documentos/Analisis# unzip -t oculto.zip
Archive:  oculto.zip
    testing: Texto.docx                   OK
No errors detected in compressed data of oculto.zip.
```

Descomprimimos con unzip oculto.zip y comprobamos que es un fichero de MS Word comprimido en un zip. ¡Hemos encontrado el contenido oculto de la primera imagen!

Ahora vamos a por la segunda, la llamada lindogatito.jpg que stegdetect nos ha dicho que lleva oculta añadida al final otra imagen.

Procedemos de la siguiente manera que con la primera usando hexeditor:

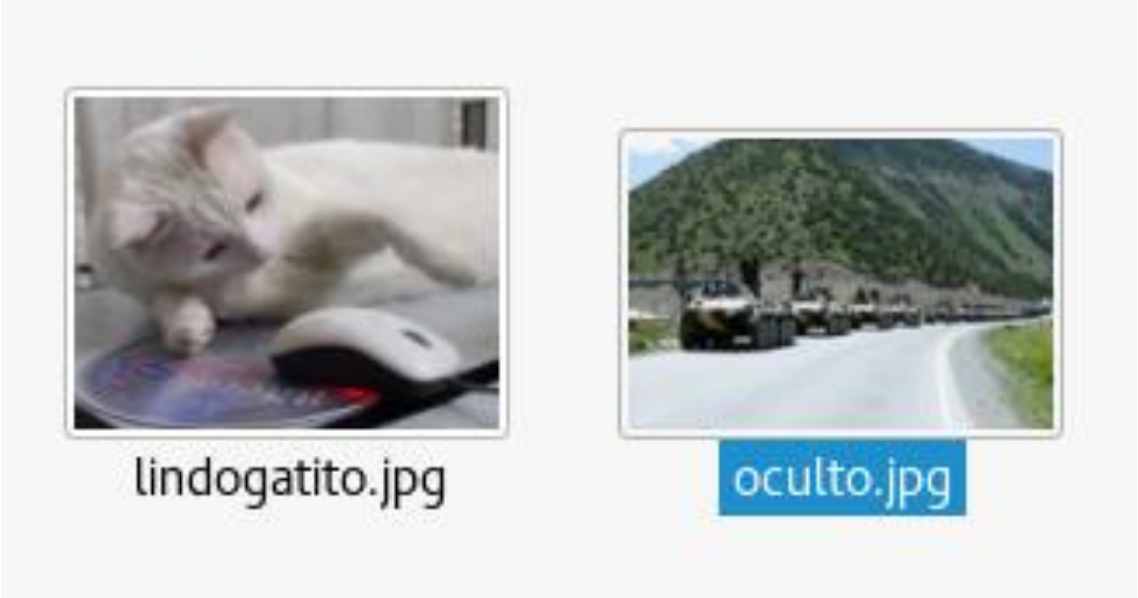
```
File: lindogatito.jpg              ASCII Offset: 0x00014314 / 0x000321BE (%40)
00014310  DD 72 E5 07  FF D9 FF D8   FF E1 00 18  45 78 69 66   .r.........Exif
00014320  00 00 49 49  2A 00 08 00   00 00 00 00  00 00 00 00   ..II*...........
```

Encontramos el final del jpg en la posición 0x14315 que indica que su longitud en decimal es de 82710 Bytes (14316 hex).

Procedemos con dd igual que anteriormente

```
dd if=lindogatito.jpg of=oculto.jpg bs=1 skip=82710
```

Y comprobamos que, efectivamente, había una imagen oculta tras el lindo gatito:



Como no queremos fiarnos de que detrás de esta imagen oculta haya algo mas ejecutamos de nuevo stegdetect para oculto.jpg, obteniendo un resultado negativo:

```
root@kali:~/Documentos/Analisis# stegdetect oculto.jpg
oculto.jpg : negative
```

CONCLUSIÓN

Como veis, ocultar información usando imágenes como contenedores, para por ejemplo sacar información de una empresa…y con esto no quiero dar ideas…es muy sencillo, pero también lo es detectarlo y extraer la información oculta con unos conocimientos mínimos de seguridad informática. En el caso de análisis forenses, no hay que descartar nunca que detrás de una foto inocente haya algo más…no todo es lo que parece.