Una al Mes: Silicon Valley Episodio 1 CTF Write-Up

1: Obtención del archivo con las credenciales.

Descargamos una imagen del disco de **Gilfoyle** desde cualquiera de los dos enlaces propuestos. Rápidamente pensamos en utilizar Volatilty para analizar dicha imagen. Primero descomprimimos el archivo descargado y luego pedimos a **Volatility** que nos dé información sobre la imagen, para poder escoger el mejor perfil a utilizar:

```
└─$unzip GILFOYLE-HELLDD.zip
Archive:  GILFOYLE-HELLDD.zip
  inflating: GILFOYLE-HELLDD.raw
┌─[socialkas@parrot]─[~/Downloads]
└─$volatility -f GILFOYLE-HELLDD.raw imageinfo
Volatility Foundation Volatility Framework 2.6
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : Win7SP1x64, Win7SP0x64, Win2008R2SP0x64, Win2008R2SP1x64_23418, Win2008R2SP1
SP1x64_23418
                     AS Layer1 : WindowsAMD64PagedMemory (Kernel AS)
                     AS Layer2 : FileAddressSpace (/home/socialkas/Downloads/GILFOYLE-HELLDD.raw)
                      PAE type : No PAE
                           DTB : 0x187000L
                          KDBG : 0xf800029f00a0L
          Number of Processors : 1
     Image Type (Service Pack) : 1
                KPCR for CPU 0 : 0xfffff800029f1d00L
             KUSER_SHARED_DATA : 0xfffff78000000000L
           Image date and time : 2018-09-15 09:56:27 UTC+0000
     Image local date and time : 2018-09-15 11:56:27 +0200
```

Obtención de información sobre la imagen del disco de Gilfoyle.

Según el enunciado del reto, debemos encontrar un archivo con las credenciales (**encriptadas**) de acceso a la web http://34.247.69.86/siliconvalley/episodio1/login.php . Como se comenta que se ha producido un corte eléctrico, podríamos pensar que tal vez ese documento estuviera en uso en el momento del corte. Así que empezamos enumerando procesos que estuvieran en ejecución para luego filtrar aquellos que puedan ser más interesantes (navegadores web, procesadores de texto, programas de anillo de claves, etc.). Para ello usamos Volatility con el primer perfil propuesto en el comando anterior:

volatility -f GILFOYLE-HELLDD.raw –profile=Win7SP1x64 pslist

Entre todos los procesos listados, observamos que **LibreOffice** y **Firefox** estaban en ejecución:

```
└─$volatility -f GILFOYLE-HELLDD.raw --profile=Win7SP1x64 pslist |grep "soffice\|firefox"
Volatility Foundation Volatility Framework 2.6
0xfffffa8002d24b30 soffice.exe           1756   1900      1       66      1      1 2018-09-15 09:48:13 UTC+0

0xfffffa8002fc7b30 soffice.bin           2340   1756     11      464      1      1 2018-09-15 09:48:18 UTC+0

0xfffffa8001d61b30 firefox.exe            956   3052      0 --------      1      0 2018-09-15 09:55:59 UTC+0
8-09-15 09:56:08 UTC+0000
```

Procesos interesantes ejecutados por el usuario.

Empezamos con LibreOffice; usamos de nuevo Volatility para listar aquellos archivos abiertos por el proceso con **PID 2340** que estuvieran ubicados en el directorio del usuario (**Users**). El primero de la lista parece muy prometedor: **info.odt:**

```
┌─[socialkas@parrot]─[~/Downloads]
└─$volatility -f GILFOYLE-HELLDD.raw --profile=Win7SP1x64 -p 2340 handles |grep "Users"
Volatility Foundation Volatility Framework 2.6
0xfffffa8001aabd50   2340             0x2e8           0x12019f File            \Device\HarddiskVolume2\Users
s\Desktop\info.odt
0xfffffa8003139d20   2340             0x354           0x12019f File            \Device\HarddiskVolume2\Users
s\AppData\Roaming\OpenOffice\4\user\uno_packages\cache\log.txt
0xfffffa80030fb730   2340             0x360           0x12019f File            \Device\HarddiskVolume2\Users
s\AppData\Roaming\OpenOffice\4\user\uno_packages\cache\uno_packages.pmap
0xfffffa8002f6ff20   2340             0x364           0x12019f File            \Device\HarddiskVolume2\Users
s\AppData\Roaming\OpenOffice\4\user\extensions\tmp\extensions.pmap
0xfffffa8002ecc3b0   2340             0x398           0x12019f File            \Device\HarddiskVolume2\Users
s\AppData\Local\Microsoft\Windows\Temporary Internet Files\counters.dat
0xfffffa8002a62410   2340             0x658           0x12019f File            \Device\HarddiskVolume2\Users
s\AppData\Local\Temp\svv5k.tmp\sv8qmu.tmp
```

Volcamos el archivo **info.odt** a disco para analizarlo con mayor detenimiento:

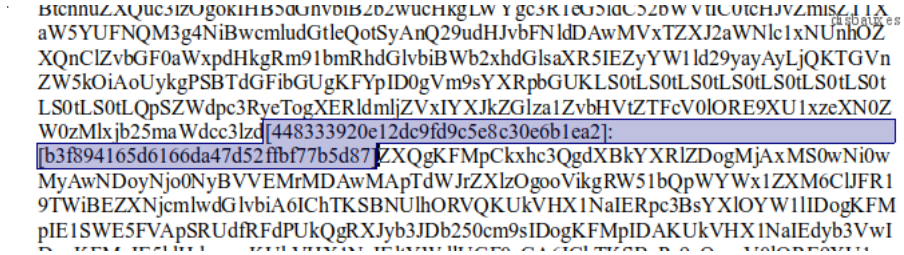
```
└─── $volatility -f GILFOYLE-HELLDD.raw --profile=Win7SP1x64 -p 2340 dumpfiles -r info.odt --dump-dir.
Volatility Foundation Volatility Framework 2.6
DataSectionObject 0xfffffa8001aabd50   2340   \Device\HarddiskVolume2\Users\unaalmes\Desktop\info.odt
```

Volcado del archivo info.odt a disco.

Abrimos el documento con LibreOffice y observamos que, efectivamente, podría muy bien ser el documento con las credenciales cifradas.

2: Descifrando las credenciales

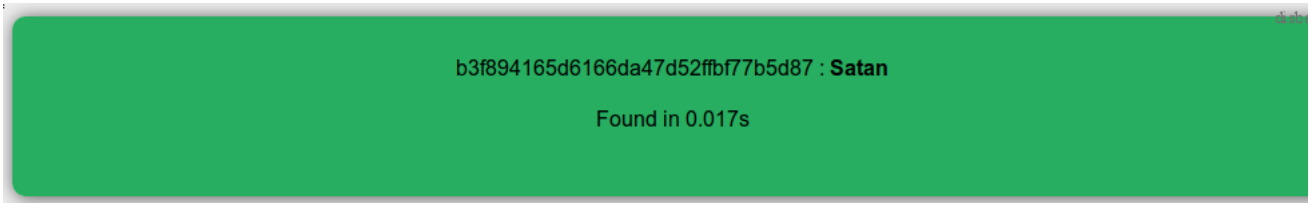
A priori, el contenido del documento parecer utilizar el abecedario de Base-64. Intentamos decodificarlo, pero sin éxito. Lo que sorprende es que el documento parece estar segmentado en grandes bloques de texto independientes, y se puede ver claramente una repetición de dichos bloques. En una parte del documento observamos una separación de párrafo después del carácter ":". Justo antes y después de dicho caracter observamos lo que podría ser un par de MD5s encerrados entre corchetes:



Posibles MD5s encerrados entre corchetes.

Lo que obtenemos bien podría ser el usuario y el password generado con MD5 que estamos buscando. Probamos suerte directamente pegando estos valores en la web http://md5decrypt.net:

Rompiendo el MD5 (usuario de la web).

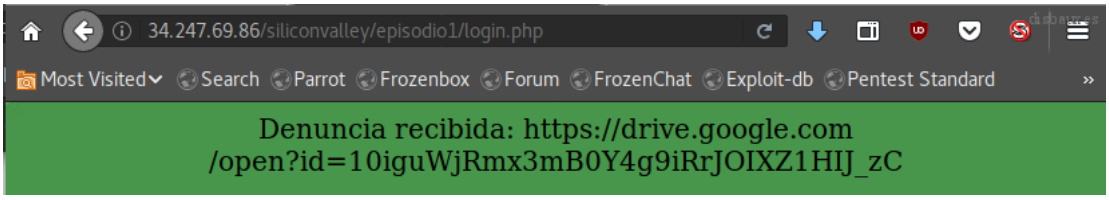


Rompiendo el MD5 (password de la web).

Ya tenemos las credenciales; podemos validarnos en la web y proseguir con el reto.

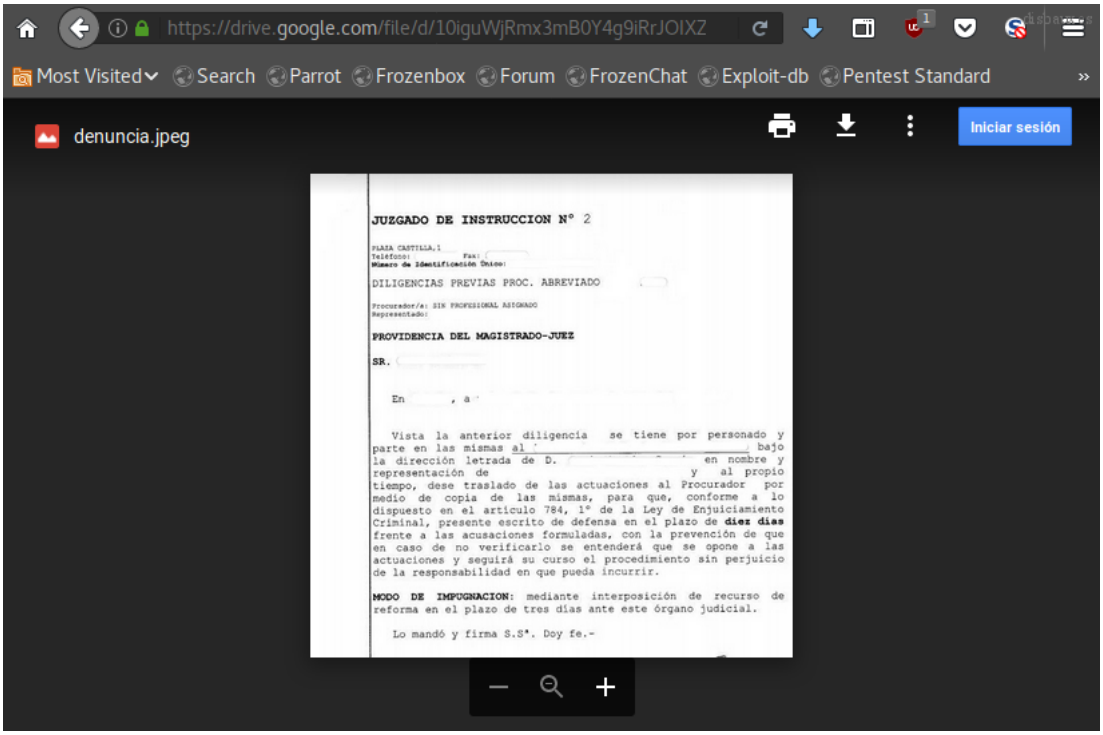
3: Obtención del número de la casa

Tras validarnos en la web con las credenciales **Gilfoyle:Satan**, obtenemos una nueva URL:



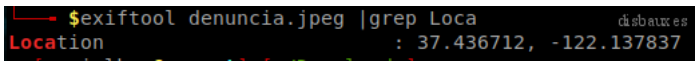
Nueva URL tras validarnos con éxito en la web.

Accedemos a la misma y obtenemos una imagen **JPG:**

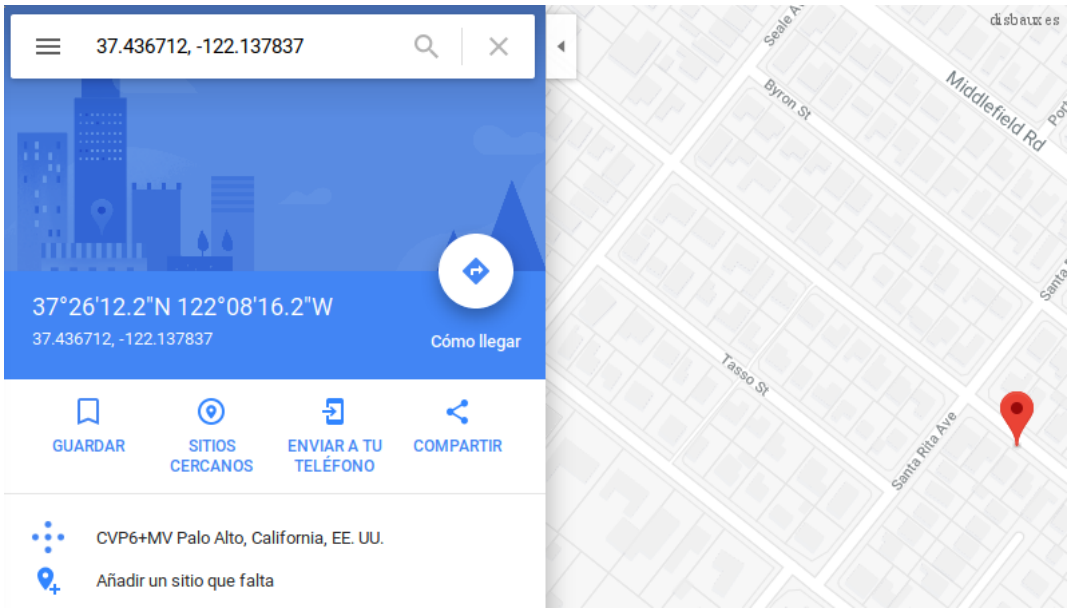


La nueva URL nos lleva hasta una imagen JPEG.

Descargamos la imagen a disco. Antes de proceder con técnicas de stego, obtenemos toda la meta-información posible sobre la imagen. Nos da por pensar: *"ey, esto parece una fotografía tomada del archivo de denuncia, tal vez tengamos suerte y tengamos la geolocalización de la cámara donde se tomó la misma…";* así que nos centramos en el campo XMP/IPTC de la imagen y buscamos la localización:

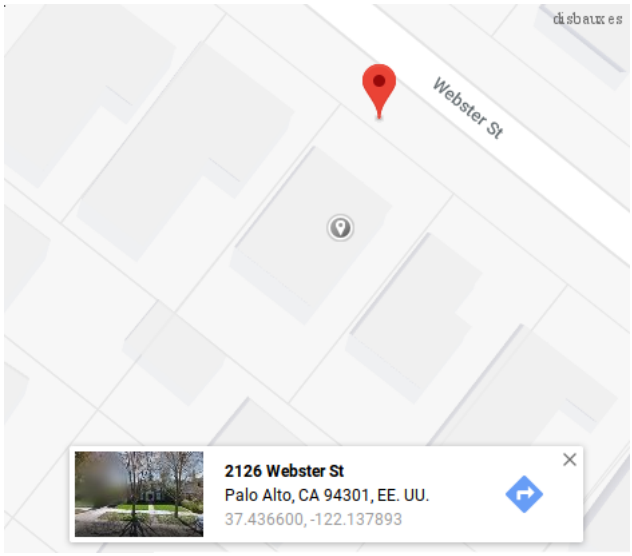


Copiamos y pegamos estas coordenadas en **Google Maps**. En lugar de un número de casa, Google nos devuelve un código **Plus** (https://plus.codes/):



Casi lo tenemos; en lugar de un número, Google nos devuelve un Plus Code.

Basta con pulsar sobre la casa que está en la calle indicada por el código **Google Plus** para obtener su número, el **2126**:



Ya tenemos el posible número de la casa, el 2126.

Generamos el **MD5** de dicho número:

echo -n "2126"|md5sum
3b92d18aa7a6176dd37d372bc2f1eb71 –

La flag es, pues: **UAM{3b92d18aa7a6176dd37d372bc2f1eb71}**

@disbauxes

Toni Castillo Girona