Módulo II: Forense

Ismael Gómez, Inés Martín y Carlos Barahona

Universidad
Rey Juan Carlos

Universidad Rey Juan Carlos

Índice

I - Análisis de RAM

Análisis de RAM

Análisis de memoria volátil

Sólo tiene contenido cuándo está conectada a la corriente y cuando se apaga el ordenador, Ciao datos.

Se almacenan de forma temporal todos los programas, procesos, librerías, etc...

I. Volatility- ¿Qué es?

¿Qué es Volatility?

Es una colección de herramientas que nos ayudan a analizar **"dumps" de memoria volátil** (**RAM**)

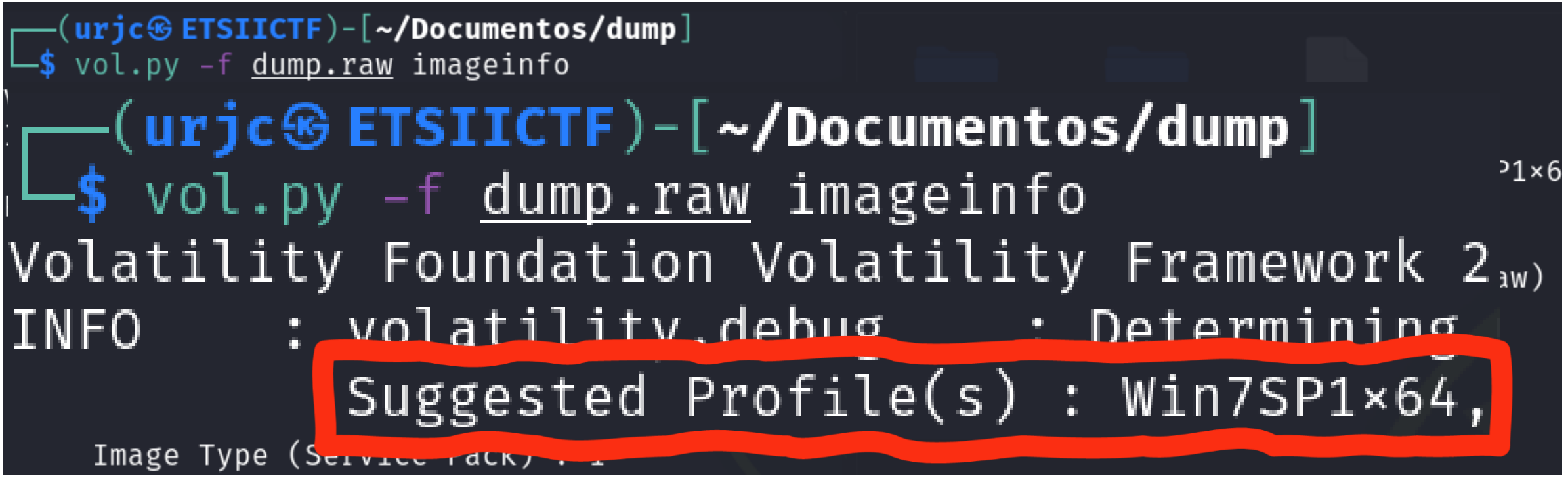
Fácil de ejecutar ya que está implementada en Python

Preinstalada en la máquina del curso

```
$ cd Documentos
$ cd volatility
$ python2 vol.py
```

Universidad Rey Juan Carlos

I. Volatility – Comandos Básicos (imageinfo)



El plugin "imageinfo" nos da información sobre el dump que vamos a comenzar a analizar

Lo más importante es quedarnos con el "profile"

II. Volatility (help)

Python2 vol.py -h

o

https://github.com/volatilityfoundation/volatility/wiki/Command-Reference

Universidad Rey Juan Carlos

I.Volatility – Comandos Básicos (pslist)

```
┌──(urjc㉿ETSIICTF)-[~/Documentos/dump]
└─$ vol.py -f dump.raw --profile="Win7SP1x64" pslist
```

Offset(V)	Name	PID	PPID	Thds	Hnds	Sess	Wow64	Start
0xfffffa801afe1b30	firefox.exe	3312	3692	33	353	1	1	2020-06-12 16:16:16 UTC+0000
0xfffffa801a811520	firefox.exe	3084	3692	39	381	1	1	2020-06-12 16:16:16 UTC+0000
0xfffffa801af39b30	firefox.exe	2784	3692	25	307	1	1	2020-06-12 16:16:21 UTC+0000
0xfffffa801aa10270	notepad.exe	3060	1928	2	58	1	0	2020-06-12 16:16:34 UTC+0000
0xfffffa8019dc1b30	sppsvc.exe	3000	512	5	164	0	0	2020-06-12 16:17:13 UTC+0000
0xfffffa801aff97d0	svchost.exe	3656	512	13	351	0	0	2020-06-12 16:17:13 UTC+0000
0xfffffa8018faf630	7zFM.exe	868	1184	4	149	1	0	2020-06-12 16:17:32 UTC+0000
0xfffffa8018f7e060	SearchProtocol	2256	1036	8	287	1	0	2020-06-12 16:18:24 UTC+0000
0xfffffa801ace08a0	SearchFilterHo	2320	1036	6	103	0	0	2020-06-12 16:18:24 UTC+0000
0xfffffa801a9d5b30	SearchProtocol	1960	1036	8	284	0	0	2020-06-12 16:18:24 UTC+0000
0xfffffa8019011b30	MRCv120.exe	1376	1928	16	319	1	1	2020-06-12 16:18:50 UTC+0000
0xfffffa8019096060	WMIADAP.exe	1184	888	6	98	0	0	2020-06-12 16:19:13 UTC+0000
0xfffffa8019066060	WmiPrvSE.exe	1400	648	8	126	0	0	2020-06-12 16:19:13 UTC+0000

Universidad Rey Juan Carlos

I. Volatility – Comandos básicos (pstree)

```
~/Desktop/retos/forense volatility -f imagen.vmem --profile=WinXPSP2x86 pstree
Volatility Foundation Volatility Framework 2.6.1
Name                                                  Pid   PPid   Thds   Hnds Time
-------------------------------------------------- ------ ------ ------ ------ ----
 0x819cc830:System                                      4      0     55    162 1970-01-01 00:00:00 UTC+0000
. 0x81945020:smss.exe                                 536      4      3     21 2011-10-10 17:03:56 UTC+0000
.. 0x816c6020:csrss.exe                               608    536     11    355 2011-10-10 17:03:58 UTC+0000
.. 0x813a9020:winlogon.exe                            632    536     24    533 2011-10-10 17:03:58 UTC+0000
... 0x816da020:services.exe                           676    632     16    261 2011-10-10 17:03:58 UTC+0000
.... 0x817757f0:svchost.exe                           916    676      9    217 2011-10-10 17:03:59 UTC+0000
.... 0x81772ca8:vmacthlp.exe                          832    676      1     24 2011-10-10 17:03:59 UTC+0000
.... 0x816c6da0:svchost.exe                           964    676     63   1058 2011-10-10 17:03:59 UTC+0000
..... 0x815c4da0:wscntfy.exe                         1920    964      1     27 2011-10-10 17:04:39 UTC+0000
..... 0x815e7be0:wuauclt.exe                          400    964      8    173 2011-10-10 17:04:46 UTC+0000
.... 0x8167e9d0:svchost.exe                           848    676     20    194 2011-10-10 17:03:59 UTC+0000
.... 0x81754990:VMwareService.e                      1444    676      3    145 2011-10-10 17:04:00 UTC+0000
.... 0x8136c5a0:alg.exe                              1616    676      7     99 2011-10-10 17:04:01 UTC+0000
.... 0x813aeda0:svchost.exe                          1148    676     12    187 2011-10-10 17:04:00 UTC+0000
.... 0x817937e0:spoolsv.exe                          1260    676     13    140 2011-10-10 17:04:00 UTC+0000
.... 0x815daca8:svchost.exe                          1020    676      5     58 2011-10-10 17:03:59 UTC+0000
... 0x813c4020:lsass.exe                              688    632     23    336 2011-10-10 17:03:58 UTC+0000
 0x813bcda0:explorer.exe                             1956   1884     18    322 2011-10-10 17:04:39 UTC+0000
```

Con este comando podemos listar los **procesos en forma de árbol**

Universidad Rey Juan Carlos

I. Volatility – Comandos básicos (cmdline)

```
┌──(urjc㉿ETSIICTF)-[~/Documentos/dump]
└─$ vol.py -f dump.raw --profile="Win7SP1x64" cmdline
```

```
**************************************************
svchost.exe pid:   3656
Command line : C:\Windows\System32\svchost.exe -k secsvcs
**************************************************
7zFM.exe pid:    868
Command line : "C:\Program Files\7-Zip\7zFM.exe" "C:\Users\Admin\Desktop\ficheroSecreto.7z"
**************************************************
```

Obtenemos los **comandos** que se ejecutaron en la máquina Windows

Universidad Rey Juan Carlos

I. Volatility – Comandos básicos (consoles)

```
volatility -f imagen.vmem --profile=WinXPSP2x86 consoles
```

```
C:\Documents and Settings\Administrator>sc query malware

SERVICE_NAME: malware
        TYPE               : 1  KERNEL_DRIVER
        STATE              : 4  RUNNING
                                (STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x0
```


Con este plugin encuentra **comandos** que un atacante puede haber ejecutado en **cmd.exe**

Universidad Rey Juan Carlos

I. Volatility – Comandos básicos (connscan)

```
volatility -f imagen.vmem --profile=WinXPSP2x86 connscan
```

```
Volatility Foundation Volatility Framework 2.6.1
Offset(P)  Local Address             Remote Address            Pid
---------- ------------------------ ------------------------- ---
0x01a25a50 0.0.0.0:1026             172.16.98.1:6666          1956
```

Listamos las **conexiones** que estaban en el momento de la captura

Universidad Rey Juan Carlos

I. Volatility – Comandos básicos (filescan)

```
volatility -f imagen.vmem --profile=WinXPSP2x86 filescan
```

```
Offset(P)             #Ptr   #Hnd Access Name
------------------ ------ ------ ------ ----
0x000000000156bcb0      2      1 ------ \Device\Afd\Endpoint
0x000000000156f100      1      1 ------ \Device\NamedPipe\W32TIME
0x00000000015a9a70      1      0 ------ \Device\KSENUM#00000002\{9B365890-165F-11D0-A195-0020AFD156E4}
0x00000000015ac5c8      1      1 R--rw- \Device\HarddiskVolume1\WINDOWS\WinSxS\x86_Microsoft.Windows.C
0x00000000015ac6b0      1      0 R--rw- \Device\HarddiskVolume1\WINDOWS\Media\Windows XP Startup.wav
0x00000000015ac8f0      1      0 R--r-d \Device\HarddiskVolume1\WINDOWS\WinSxS\x86_Microsoft.VC80.MFC_
0x00000000015ad318      1      0 R--r-d \Device\HarddiskVolume1\WINDOWS\system32\webcheck.dll
0x00000000015ad740      1      0 R--r-d \Device\HarddiskVolume1\WINDOWS\system32\themeui.dll
```


Con este comando podemos listar los **archivos** que se encontraban en la máquina

Universidad Rey Juan Carlos

I. Volatility – Comandos básicos (dumpfile)

```
~/Desktop/retos/forense volatility -f imagen.vmem --profile=WinXPSP2x86 filescan | grep .wav
Volatility Foundation Volatility Framework 2.6.1
0x00000000015ac6b0      1      0 R--rw- \Device\HarddiskVolume1\WINDOWS\Media\Windows XP Startup.wav
0x00000000018d82c0      1      0 R--rw- \Device\HarddiskVolume1\WINDOWS\Media\Windows XP Balloon.wav
```

```
~/Desktop/retos/forense volatility -f imagen.vmem --profile=WinXPSP2x86 dumpfiles --dump-dir=. -Q 0x00000000015ac6b0
Volatility Foundation Volatility Framework 2.6.1
DataSectionObject 0x015ac6b0   None   \Device\HarddiskVolume1\WINDOWS\Media\Windows XP Startup.wav
```


Con este comando podemos **dumpear/extraer archivos concretos** que se encontraban en la máquina

Universidad Rey Juan Carlos

I. Volatility – Comandos básicos (hashdump)

```
┌──(urjc㉿ETSIICTF)-[~/Documentos/dump]
└─$ vol.py -f dump.raw --profile="Win7SP1x64" hashdump
Volatility Foundation Volatility Framework 2.6.1
Administrador:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Invitado:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Admin:1000:aad3b435b51404eeaad3b435b51404ee:62234517c6b66dc7839f0da943bd29ee:::
```

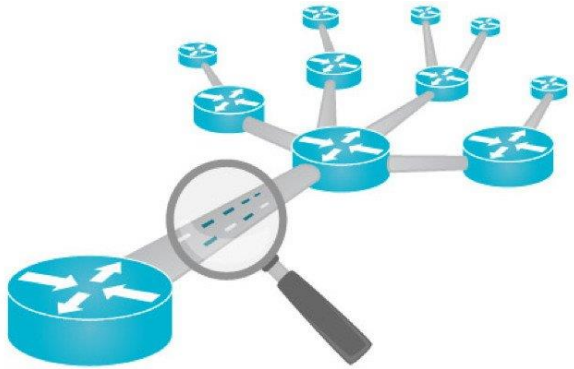

Con este comando podemos **dumpear/extraer los hashes** de los usuarios de la máquina

II – Análisis de tráfico

Análisis de tráfico

Análisis de las actividades de la red para descubrir el origen de ataques, virus, intrusiones o infracciones de seguridad que se producen en una red.

Involucra las redes informáticas y los protocolos de red.



Permitirá descubrir:

- Navegación en páginas web
- Exfiltraciones de datos
- Conexiones maliciosas
- Credenciales en texto plano
- …

II – Wireshark

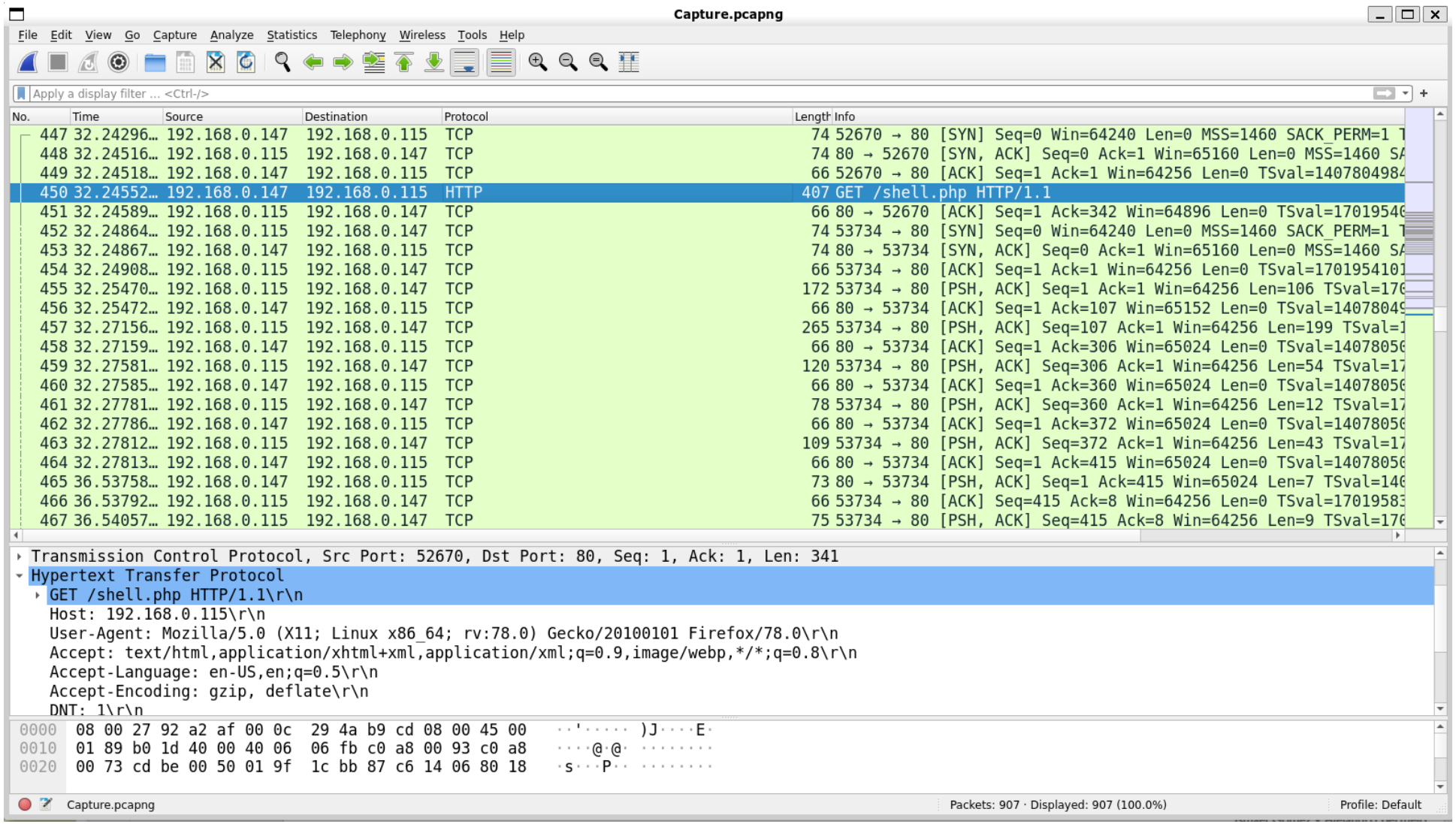
¿Qué es Wireshark?

Es un "sniffer" o herramienta que intercepta tráfico. Muestra en una interfaz sencilla paquete a paquete y todos los datos que contienen. Admite más de 2000 protocolos de red.

Las capturas de tráfico se guardan en ficheros .pcap, que es con lo que vamos a trabajar mayoritariamente en CTFs

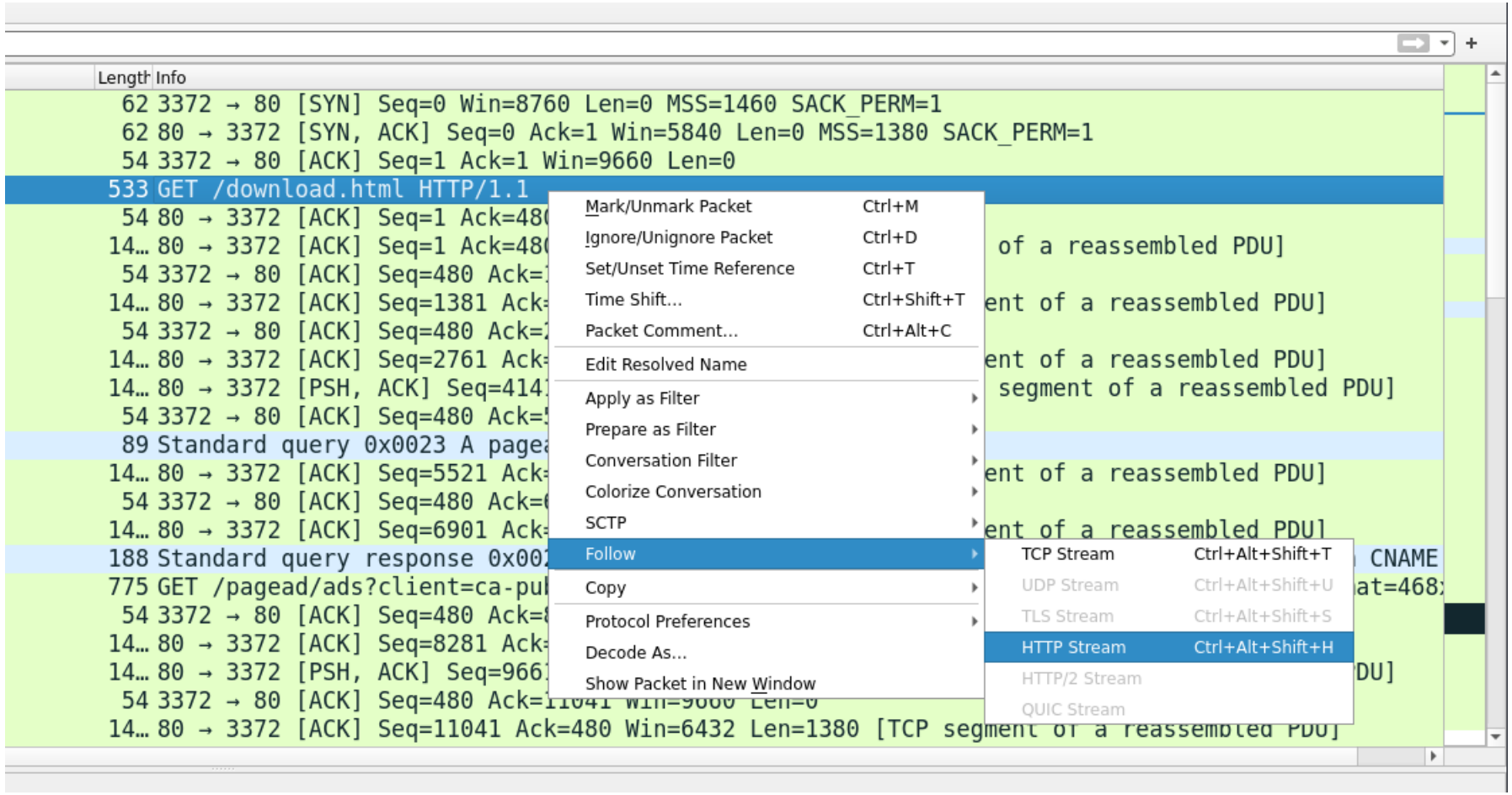
(la captura nos la dan)

II – Wireshark



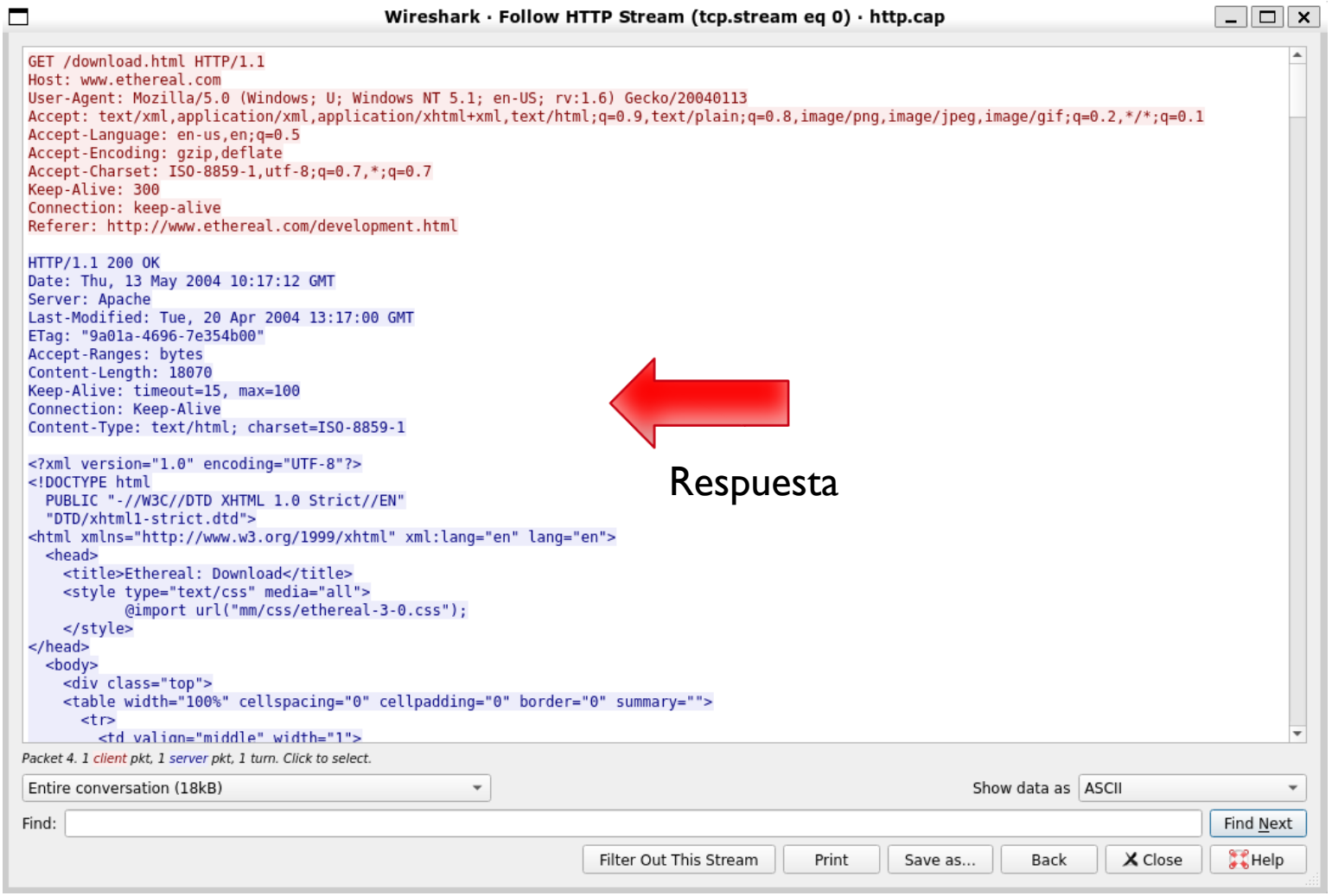
II – Wireshark *(follow stream)*

Seguir flujo HTTP



II – Wireshark *(follow stream)*

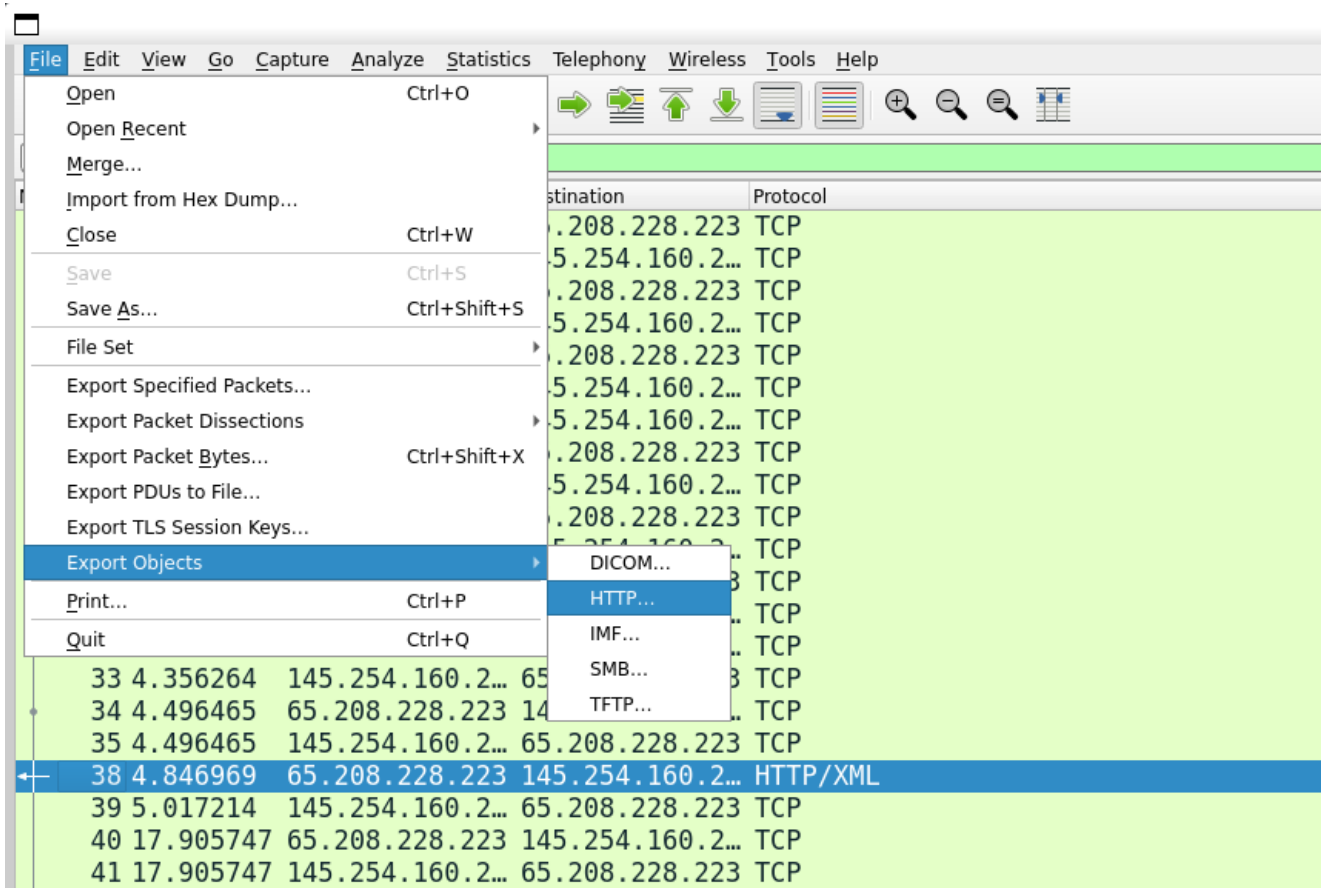
Petición



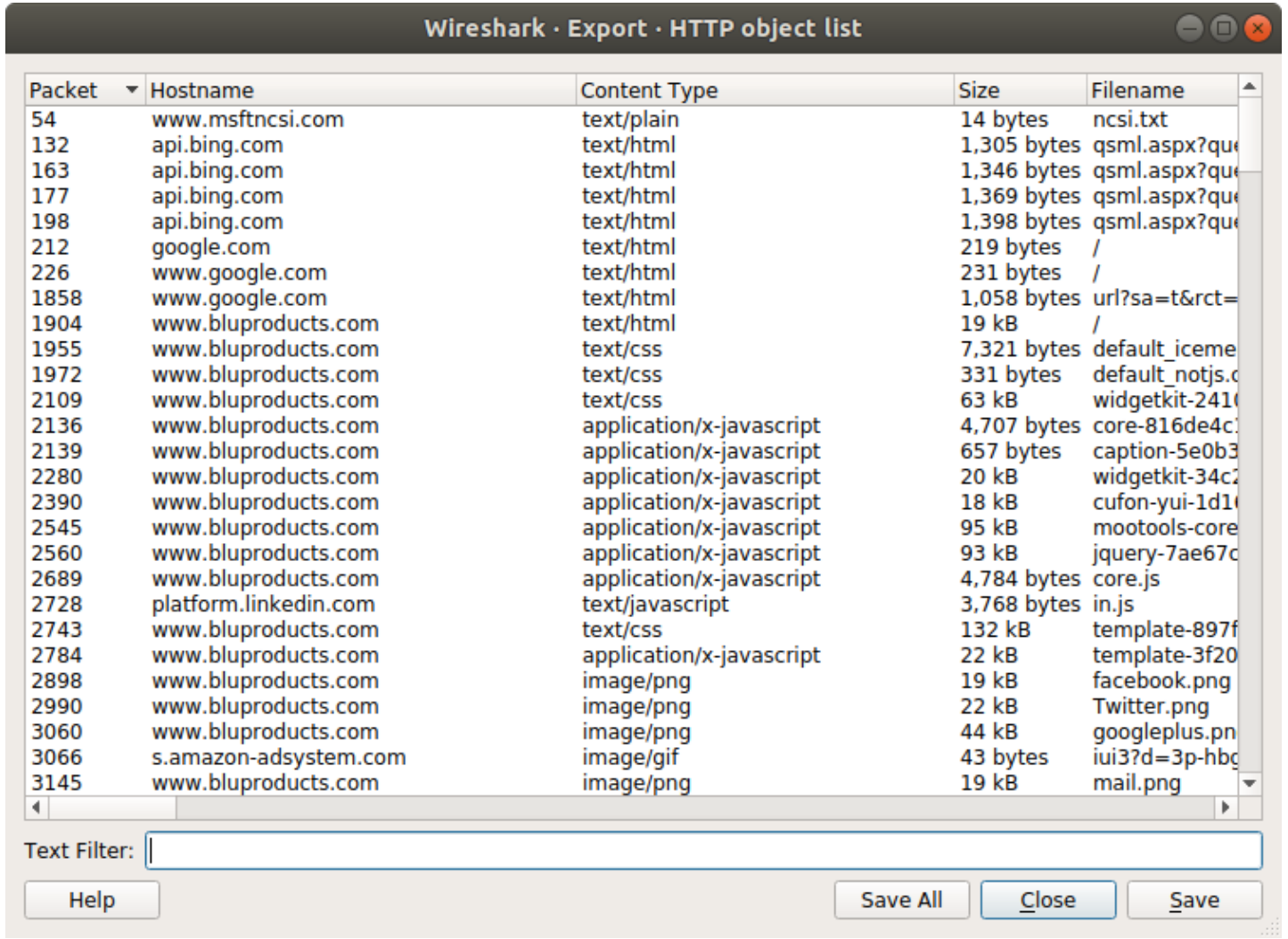
Respuesta

II – Wireshark *(export objects)*

Exportar objetos



II – Wireshark *(export objects)*



II – Wireshark *(filters)*

Filtros de Wireshark

Podemos filtrar los paquetes en base a diferentes campos:

Direcciones IP

- IP: ip.addr == 10.10.50.1
- Origen: ip.src == 10.10.50.1
- Destino: ip.dest == 10.10.50.1
- Subred: ip.addr == 10.10.50.1/24

Protocolos

- tcp
- udp
- dns
- http
- ftp
- …

Operadores

- and o &&
- or o ||
- xor o ^^
- not o !

Texto

- Edit → Find packet → String

II – Wireshark *(filters)*

Ejemplo

ftp.request && ip.src == 192.168.0.147

No.	Time	Source	Destination	Protocol	Length	Info
241	4.035759…	192.168.0.147	192.168.0.115	FTP	78	Request: USER jenny
269	4.043289…	192.168.0.147	192.168.0.115	FTP	78	Request: USER jenny
273	4.108928…	192.168.0.147	192.168.0.115	FTP	81	Request: PASS football
274	4.121641…	192.168.0.147	192.168.0.115	FTP	79	Request: PASS 000000
275	4.121775…	192.168.0.147	192.168.0.115	FTP	83	Request: PASS 1234567890
276	4.133276…	192.168.0.147	192.168.0.115	FTP	81	Request: PASS computer
277	4.139140…	192.168.0.147	192.168.0.115	FTP	81	Request: PASS superman
278	4.140089…	192.168.0.147	192.168.0.115	FTP	81	Request: PASS internet
279	4.141101…	192.168.0.147	192.168.0.115	FTP	84	Request: PASS password123
280	4.141239…	192.168.0.147	192.168.0.115	FTP	81	Request: PASS 1qaz2wsx
281	4.143016…	192.168.0.147	192.168.0.115	FTP	79	Request: PASS monkey
282	4.143070…	192.168.0.147	192.168.0.115	FTP	80	Request: PASS michael
283	4.143117…	192.168.0.147	192.168.0.115	FTP	79	Request: PASS shadow

Hemos usado dos filtros concatenados con (&&)

I. ftp.request ➜ Nos muestra todas las "request" del protocolo ftp
II. Ip.src == 192.168.0.147 ➜ Nos muestra todos los paquetes que vienen de la IP "192.168.0.147"

II – Wireshark *(protocol hierarchy)*

Jerarquía de protocolos

Estadísticas → Jerarquía de protocolo

Wireshark · Estadísticas de jerarquía de protocolo · capture.pcapng

Protocolo	Porcentaje de paquetes	Paquetes	Porcentaje de bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	85634	100.0	18726049	62k	0	0	0
Linux cooked-mode capture	100.0	85634	7.3	1370144	4600	0	0	0
Internet Protocol Version 6	0.1	86	0.0	3440	11	0	0	0
Internet Protocol Version 4	64.5	55272	5.9	1105964	3713	0	0	0
User Datagram Protocol	0.5	444	0.0	3552	11	0	0	0
Simple Service Discovery Protocol	0.0	11	0.0	3008	10	11	3008	10
QUIC IETF	0.2	166	0.5	88082	295	119	50692	170
NetBIOS Datagram Service	0.0	3	0.0	603	2	0	0	0
SMB (Server Message Block Protocol)	0.0	3	0.0	357	1	0	0	0
Domain Name System	0.4	309	0.3	58393	196	309	58393	196
Data	0.0	2	0.0	600	2	2	600	2
Transmission Control Protocol	63.9	54687	81.4	15242898	51k	24794	4335343	14k
Transport Layer Security	2.1	1815	12.7	2383885	8004	1791	2241486	7525
Malformed Packet	0.0	7	0.0	0	0	7	0	0
Hypertext Transfer Protocol	25.2	21573	45.6	8533837	28k	10312	1584477	5319
Portable Network Graphics	0.0	5	0.1	12319	41	5	12319	41
Online Certificate Status Protocol	0.0	16	0.0	4348	14	16	4348	14
Media Type	0.0	22	6.8	1264094	4244	22	902661	3030
Line-based text data	12.5	10700	30.6	5723438	19k	10700	4916620	16k
JPEG File Interchange Format	0.0	6	3.2	603766	2027	6	604920	2031
HTML Form URL Encoded	0.6	492	0.1	22856	76	492	142144	477
eXtensible Markup Language	0.0	8	0.1	20896	70	8	20896	70
Compuserve GIF	0.0	12	0.0	2486	8	12	2486	8
FTP Data	0.1	72	7.1	1330793	4468	0	0	0
File Transfer Protocol (FTP)	5.3	4573	0.5	93259	313	4573	0	0
Data	2.2	1877	0.2	39960	134	1877	39960	134
Internet Group Management Protocol	0.2	131	0.0	1048	3	131	1048	3
Internet Control Message Protocol	0.0	10	0.0	1816	6	8	1144	3
Data	0.0	2	0.0	600	2	2	600	2
Address Resolution Protocol	35.4	30276	4.6	854442	2868	30276	854442	2868

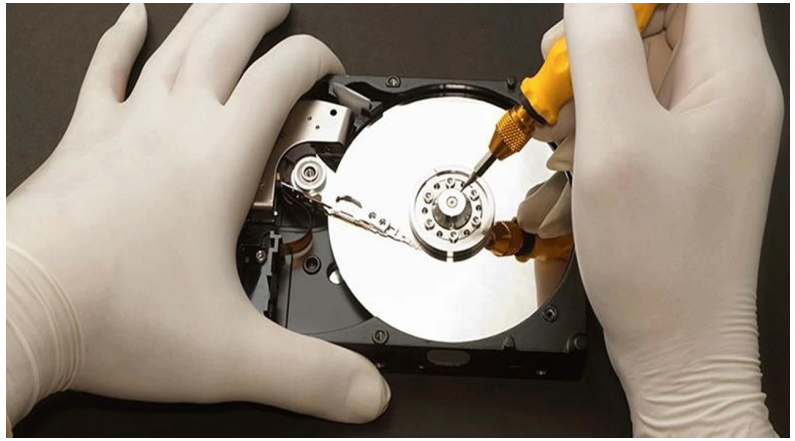
No hay filtro de visualización.

Cerrar | Copiar | Ayuda

II – Wireshark (Retos)

Análisis de discos

Extracción de información forense de los medios de almacenamiento digital, como pueden ser discos duros, dispositivos USB, unidades flash, CDs o DVDs…



III – Análisis de discos: ¿qué buscar?

Artefactos

- **Datos de navegación:** historial, cookies, credenciales guardadas…
- **Descarga de archivos:** navegador, adjuntos de emails…
- **Ejecución de programas**
- **Borrado de archivos:** papelera de reciclaje, file carving…
- **Uso de cuentas:** ¿quién fue el último usuario en loguearse? ¿qué hizo? …
- **Apertura de archivos/carpetas recientes**

…

III – Autopsy

¿Qué es Autopsy?

Autopsy es una herramienta utilizada en el ámbito forense que sirve para analizar imágenes de **disco**, tanto de Windows como de sistemas UNIX

(NTFS, Fat, Ext3/4, ….)

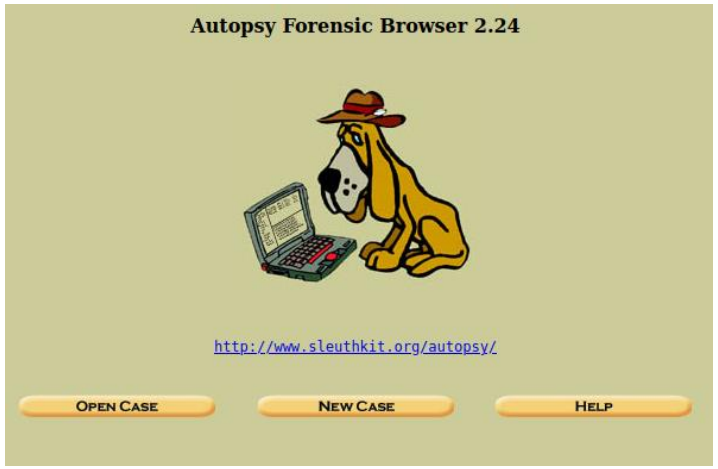
III – Autopsy: primeros pasos

Instalación y uso

Windows

https://www.autopsy.com/download/

Linux/OS X



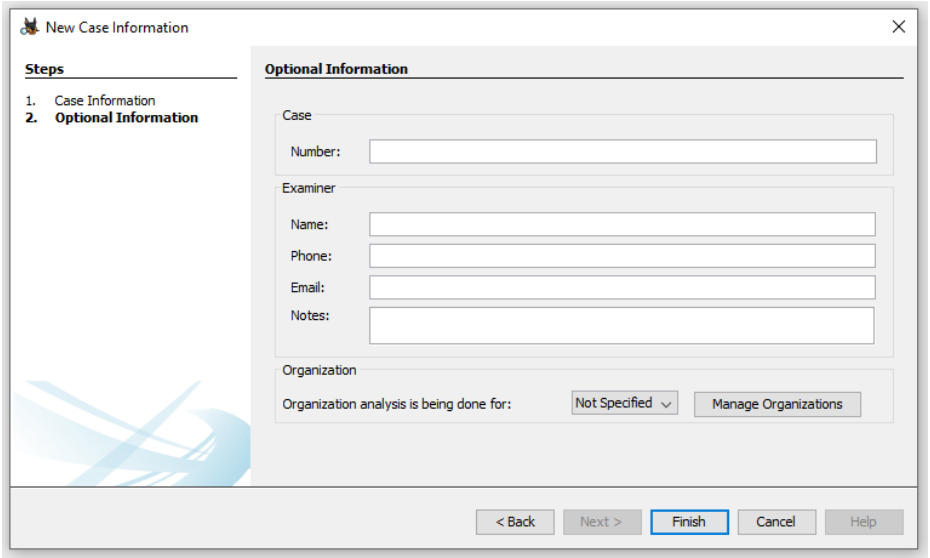
$ sudo apt-get install autopsy

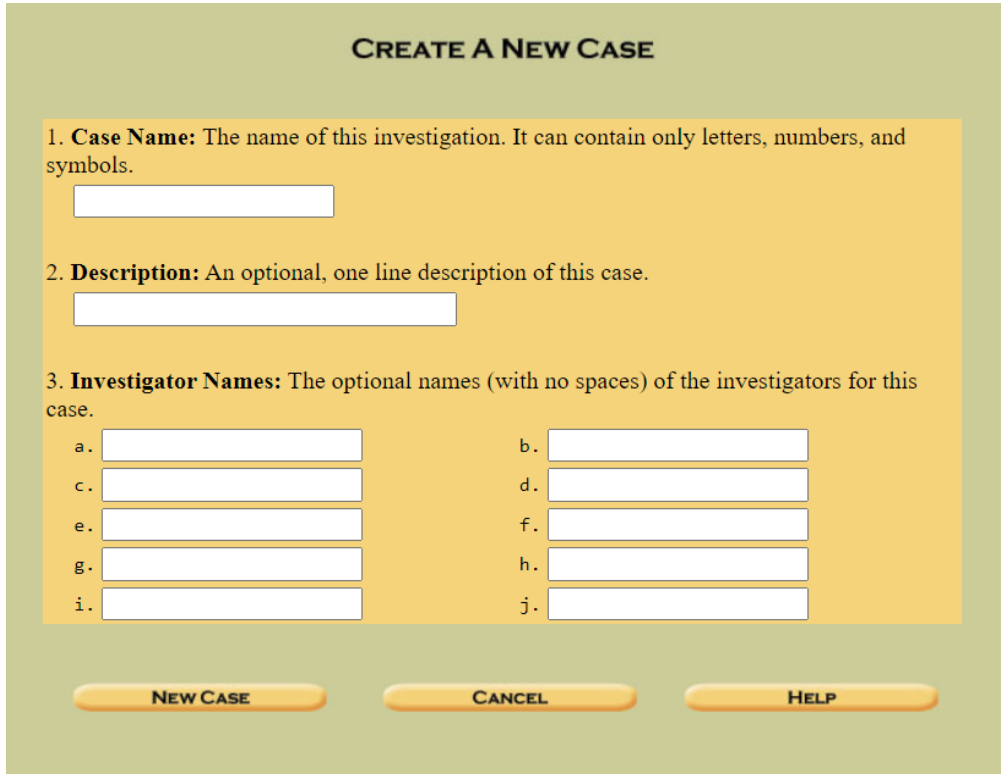
(preinstalada en la VM del curso)

III – Autopsy: primeros pasos

1. Crear un caso

Habrá que rellenar información que describa el caso que ha a analizar: nombre del caso, descripción, nombre de los investigadores…

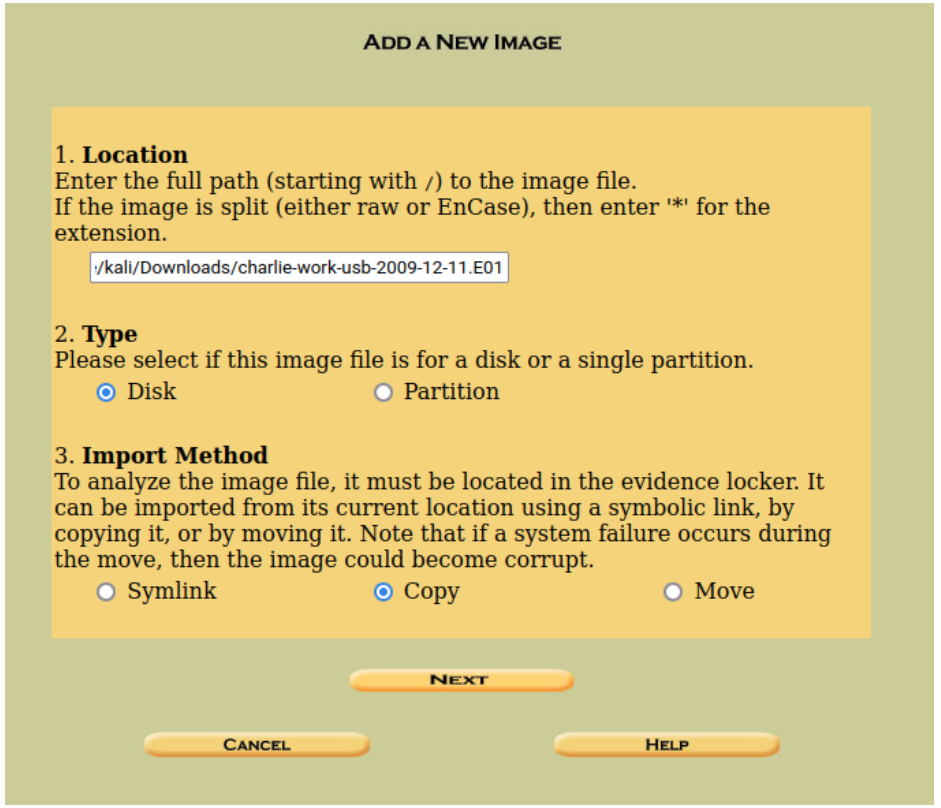




III – Autopsy: primeros pasos

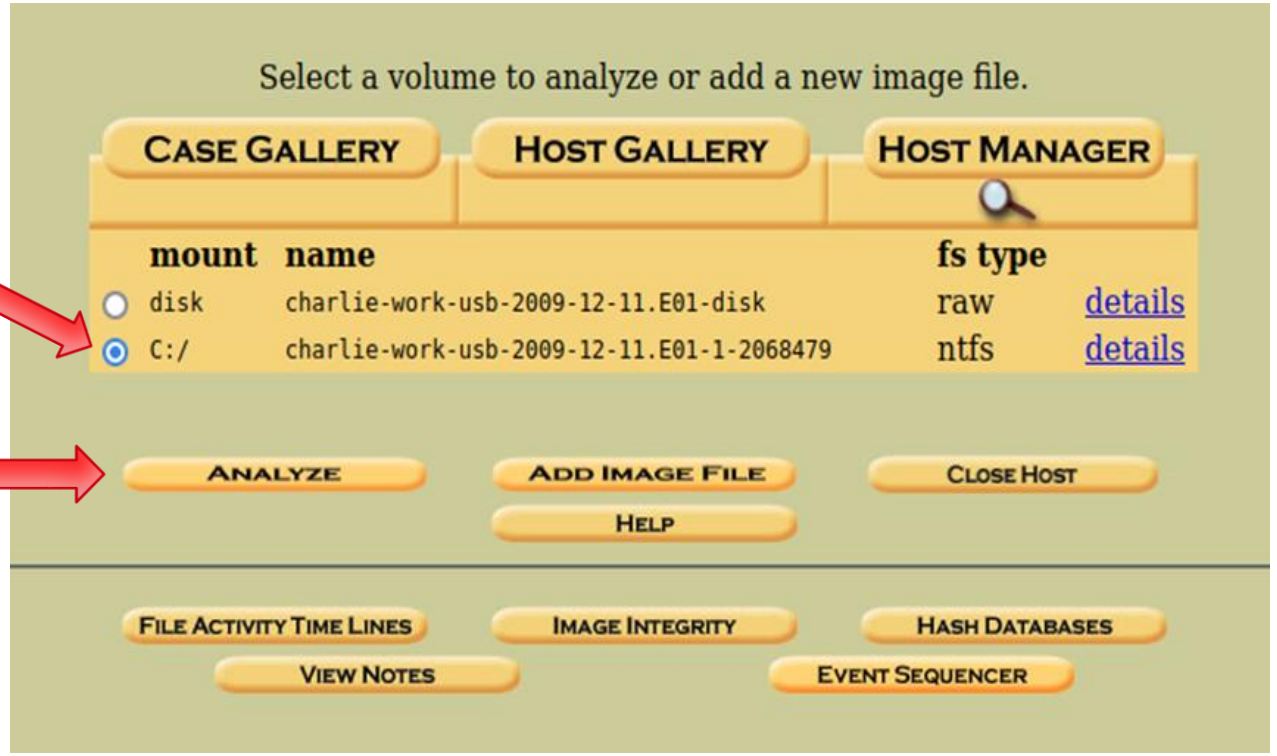
II. Agregar una imagen

- **Location:** ruta completa al fichero a analizar.
- **Type:** si el fichero es una imagen de disco entera o una partición (si no estamos seguros lo dejamos en disco)
- **Método de importación:** elegir el más conveniente



III – Autopsy: análisis del disco

III. Analizar el disco



III – Autopsy: análisis del disco

III. Tipos de análisis

- **File analysis:** análisis del sistema de ficheros.
- **Keyword search:** búsqueda de texto. Admite expresiones regulares.
- **File type:** categoriza los archivos por su extensión. Intenta además detectar aquellos archivos que tengan su extensión cambiada.
  - **Meta data:** detalles sobre entradas MFT en el sistema de archivos.
  - **Data unit:** permite ver datos de distintas formas (hexadecimal, por ejemplo).

III – Autopsy: análisis del disco

IV. Análisis

Directory Seek

Enter the name of a directory that you want to view.

C:/ [ ] VIEW

File Name Search

Enter a Perl regular expression for the file names you want to find.

[ ] SEARCH

ALL DELETED FILES

EXPAND DIRECTORIES

Current Directory: C:/ /Email/

ADD NOTE | GENERATE MD5 LIST OF FILES

DEL	Type dir / in	NAME	WRITTEN	ACCESSED	CHANGED	CREATED	SIZE
	d / d	../	2009-12-03 16:17:01 (EST)	2009-12-10 17:26:03 (EST)	2009-12-03 16:17:01 (EST)	2009-11-20 12:38:09 (EST)	56
	d / d	./	2009-12-10 17:27:55 (EST)	2009-12-10 17:29:38 (EST)	2009-12-10 17:27:55 (EST)	2009-12-03 16:16:59 (EST)	56
	d / d	other/	2009-12-10 17:39:09 (EST)	2009-12-10 17:39:09 (EST)	2009-12-10 17:39:09 (EST)	2009-12-10 17:27:49 (EST)	56
	r / r	Charlie_2009-11-16_1102_Received.txt	2009-12-03 16:19:12 (EST)	2009-12-04 16:52:39 (EST)	2009-12-04 12:14:21 (EST)	2009-12-03 16:19:12 (EST)	676

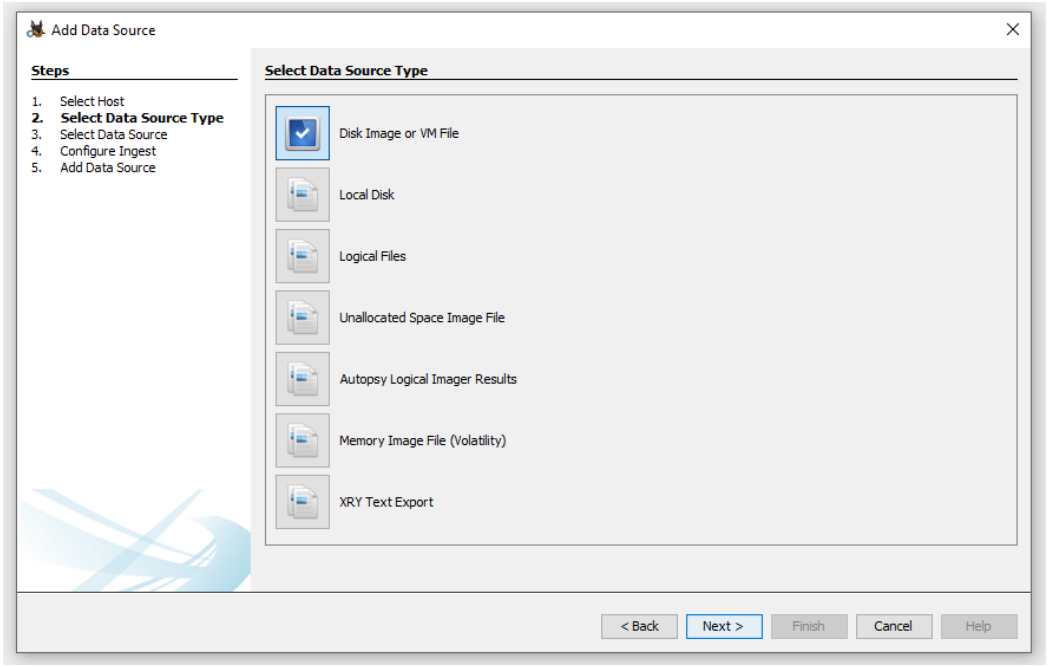
ASCII (display - report) * Hex (display - report) * ASCII Strings (display - report) * Export * Add Note
File Type: ASCII text, with CRLF line terminators

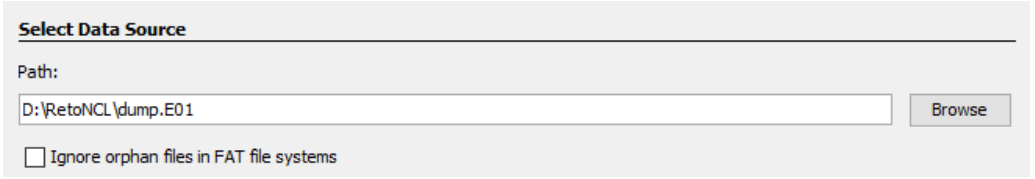
```
Contents Of File: C:/Email/Charlie_2009-11-16_1102_Received.txt


Subject:
WELCOME TO THE COMPANY!
From:
"Pat McGoo" <pat@m57.biz>
Date:
Mon, 16 Nov 2009 11:02:37 -0800
To:
<charlie@m57.biz>, <jo@m57.biz>, <terry@m57.biz>

Dear Team,

I am extremely excited to take this opportunity to welcome you all to the M57.biz family.  It has been a dream of mine to open a business that provides an innovative service to companies, inventors,

I look forward to all of your great work in your future assignments and I can't wait to get to know each of you a little more.  Please feel free to send me any questions, concerns, or comments.

Regards,
```

III – Autopsy: primeros pasos

II. Agregar una imagen

- Elegir tipo de imagen
- Elegir la ruta del archivo



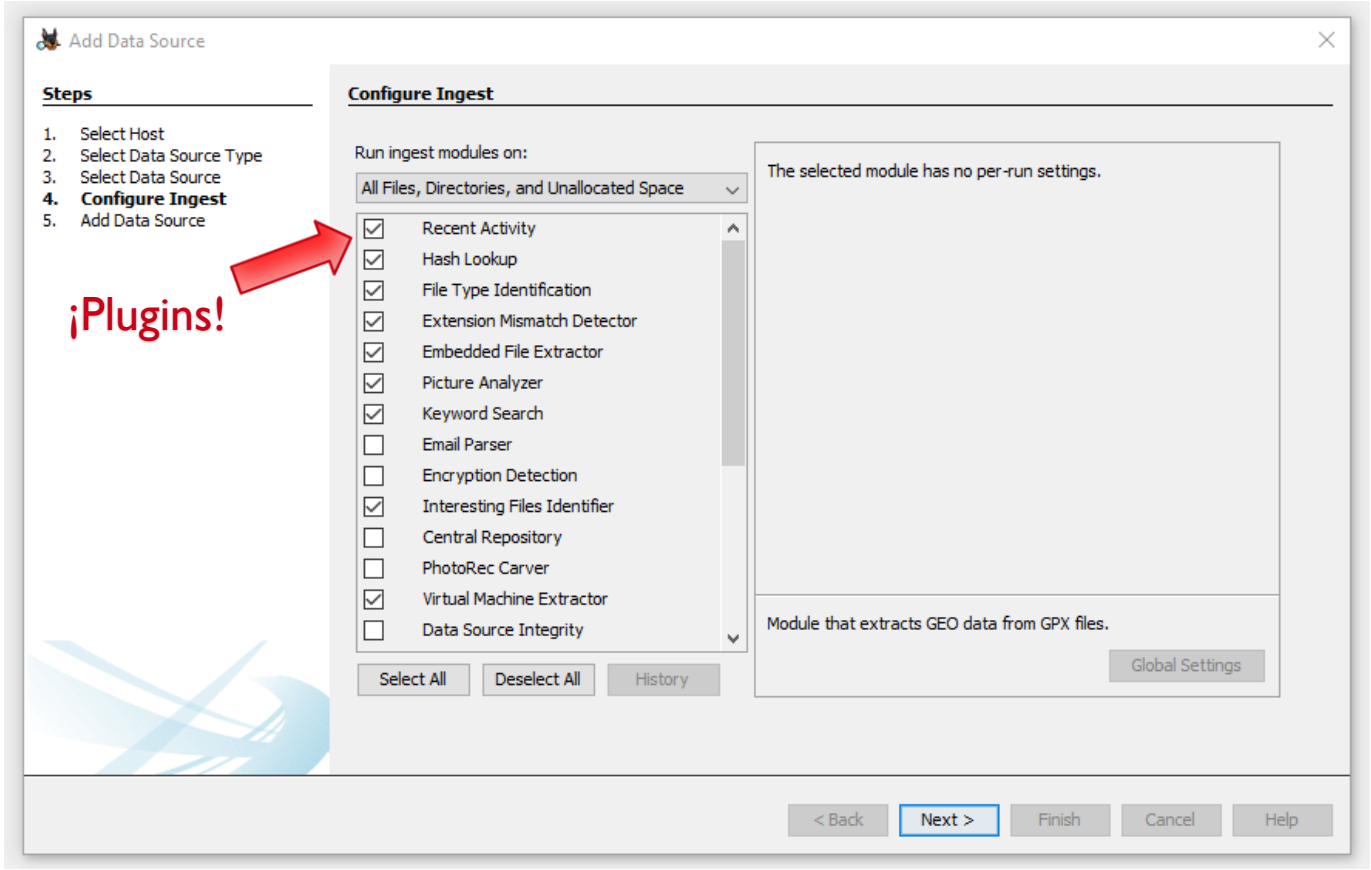


Tipos de imágenes soportadas

- **Disk Image/VM file:** imágenes que son una copia exacta de un disco duro, o una máquina virtual.
- **Local disk:** disco duro, pendrive, tarjeta de memoria…
- **Logical files:** carpetas o archivos locales.
- **Unallocated space image file:** espacio no asignado.

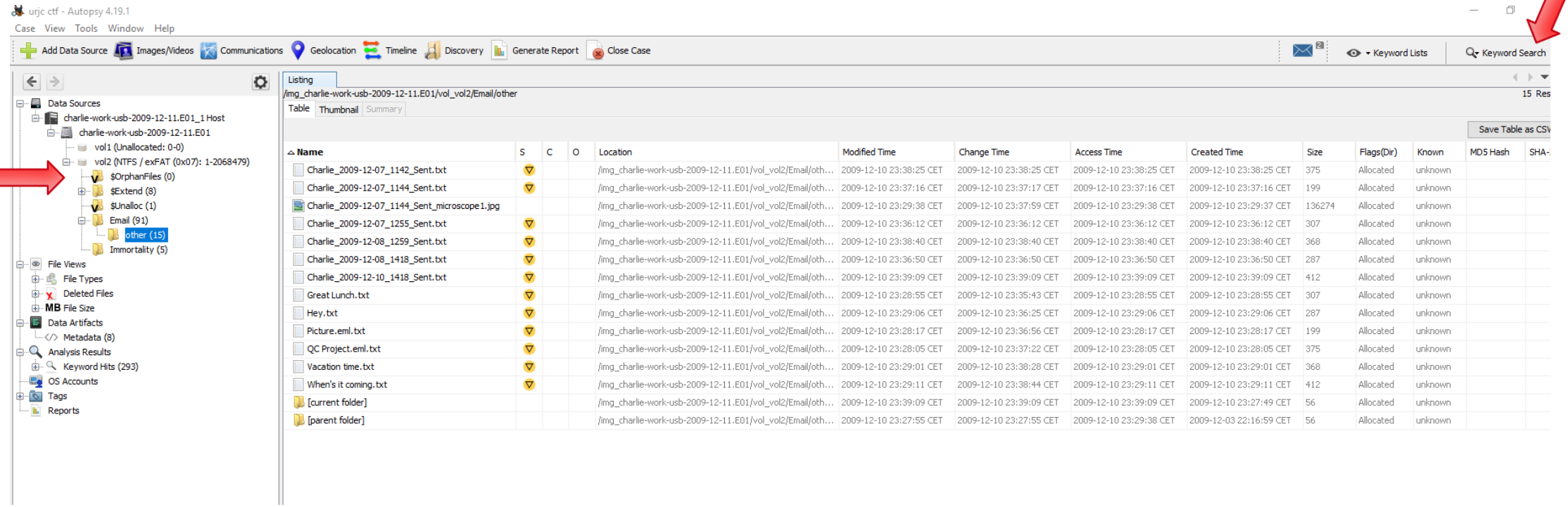
III – Autopsy: análisis del disco

III. Tipos de análisis



III – Autopsy: análisis del disco

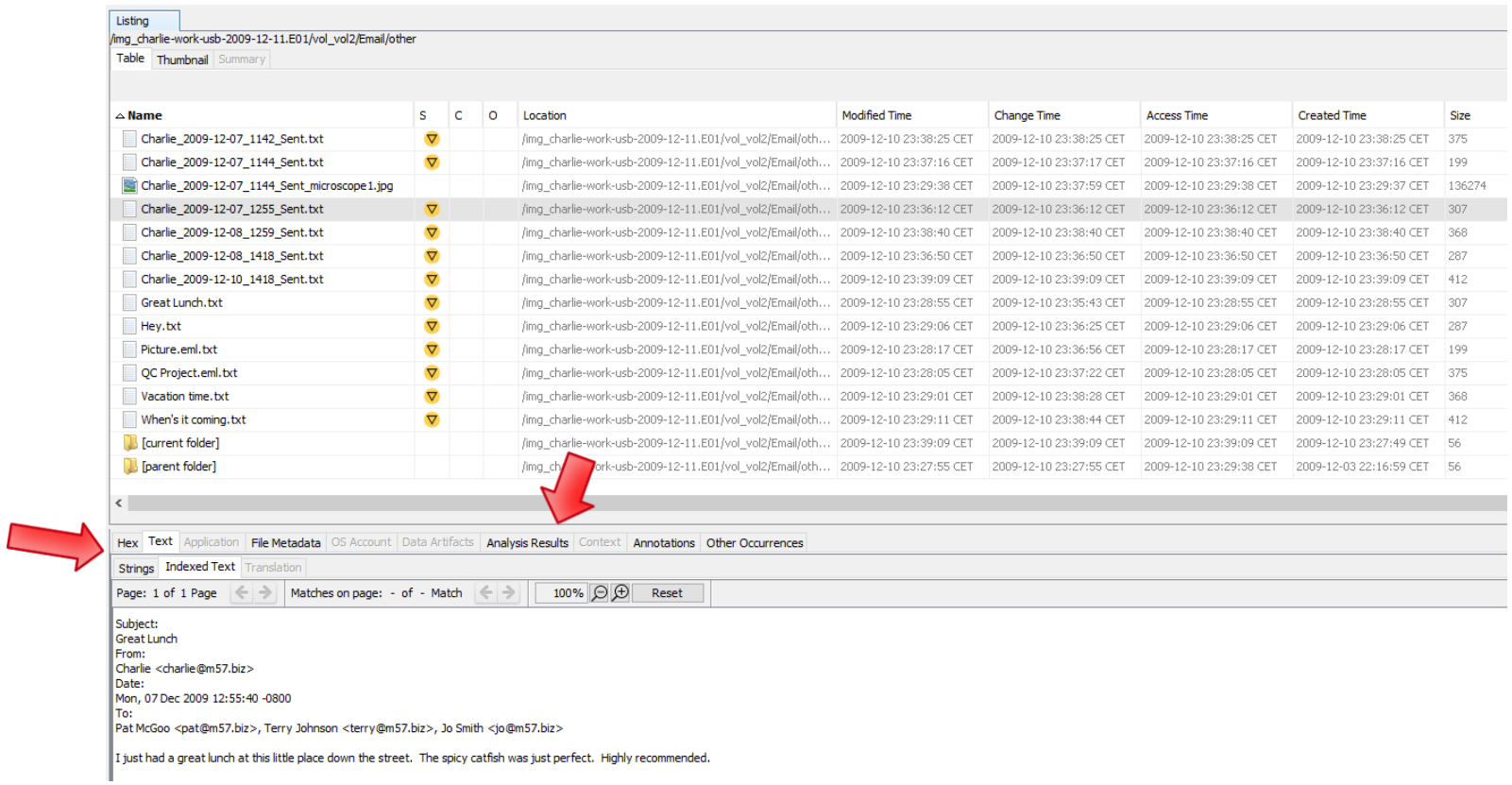
IV. Analizar el disco



Información

III – Autopsy: análisis del disco

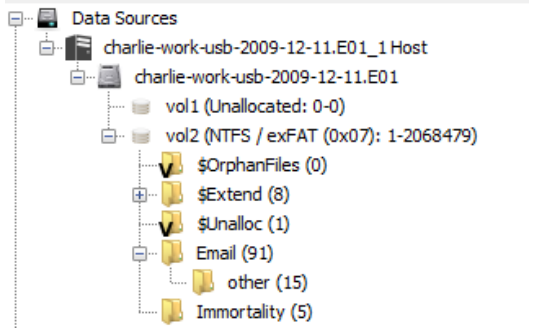
IV. Analizar el disco



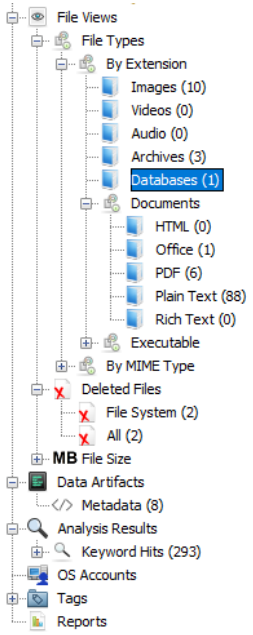
III – Autopsy: análisis del disco

IV. Analizar el disco

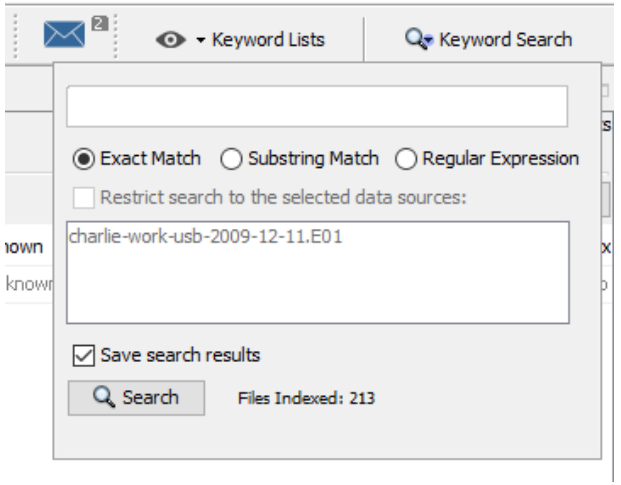
Navegación por directorios



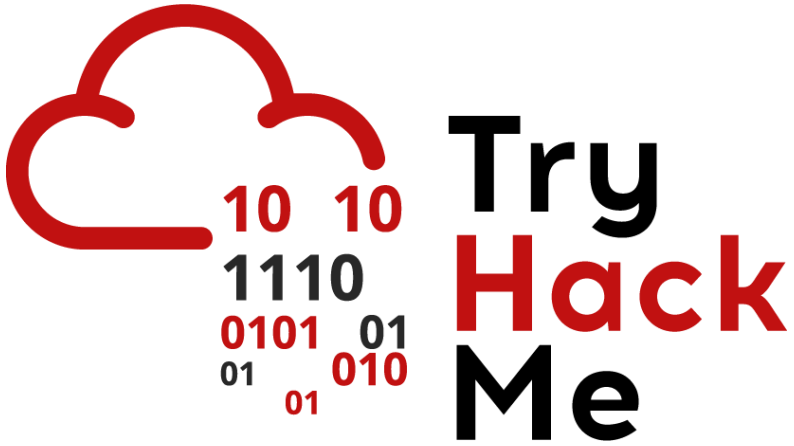
Categorización de archivos
+ Información relevante



Búsqueda de texto



III – Autopsy: para practicar



Disk Analysis & Autopsy

Módulo II: Forense

Ismael Gómez, Inés Martín y Carlos Barahona

Universidad
Rey Juan Carlos