

Modulo II: Forense

Ismael Gómez Esquilichi y Alejandro Bermejo Pérez



Universidad
Rey Juan Carlos

Índice

1. Volatility

1. ¿Qué es?
2. Comandos básicos
3. Dumpeo de archivos

2. Wireshark

1. ¿Qué es?
2. Ejemplos de uso

3. Autopsy

1. ¿Qué es?
2. Abrir un caso
3. Tipos de análisis

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



I. Volatility – Comandos Básicos (imageinfo)

```
(urjc@ETSIICTF)-[~/Documentos/dump]
$ vol.py -f dump.raw imageinfo


(urjc@ETSIICTF)-[~/Documentos/dump]
$ vol.py -f dump.raw imageinfo
Volatility Foundation Volatility Framework 2.4.0
INFO : volatility.debug : Determining
Suggested Profile(s) : Win7SP1x64,
Image Type (Service Pack) : 1
```

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

Lo más importante es quedarnos con el "profile"

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	spsvc.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

I. Volatility – Comandos básicos (pstree)

```
Apple > ~/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:wuauc.lt.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

I. Volatility – Comandos básicos (cmdline)

```
(urjc@ETSICTF)-[~/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

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**

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

I. Volatility – Comandos básicos (filescan)

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

Offset(P)	#Ptr	#Hnd	Access	Name
0x00000000156bcb0	2	1	-----	\Device\Afd\Endpoint
0x00000000156f100	1	1	-----	\Device\NamedPipe\W32TIME
0x0000000015a9a70	1	0	-----	\Device\KSENUM#00000002\{9B365890-165F-11D0-A195-0020AFD156E4}
0x0000000015ac5c8	1	1	R--rw-	\Device\HarddiskVolume1\WINDOWS\WinSxS\x86_Microsoft.Windows.C
0x0000000015ac6b0	1	0	R--rw-	\Device\HarddiskVolume1\WINDOWS\Media\Windows XP Startup.wav
0x0000000015ac8f0	1	0	R--r-d	\Device\HarddiskVolume1\WINDOWS\WinSxS\x86_Microsoft.VC80.MFC_
0x0000000015ad318	1	0	R--r-d	\Device\HarddiskVolume1\WINDOWS\system32\webcheck.dll
0x0000000015ad740	1	0	R--r-d	\Device\HarddiskVolume1\WINDOWS\system32\themeui.dll

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

I. Volatility – Comandos básicos (dumpfile)

```
Apple > ~/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
```

```
Apple > ~/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

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 - Wireshark

¿Qué es Wireshark?

Es una herramienta que intercepta tráfico/sniffer (admite más de 2000 protocolos de red), que muestra en una interfaz sencilla paquete a paquete y todos los datos que contiene..

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

The image shows a Wireshark network traffic capture window titled "Capture.pcapng". The main pane displays a list of captured packets. A red arrow points to packet 450, which is an HTTP GET request for "/shell.php". The packet details pane on the right shows the structure of the request, including the Hypertext Transfer Protocol section. A second red arrow points to the "GET /shell.php HTTP/1.1\r\n" line in the details pane. The packet bytes pane at the bottom shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
447	32.24296...	192.168.0.147	192.168.0.115	TCP	74	52670 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 T...
448	32.24516...	192.168.0.115	192.168.0.147	TCP	74	80 → 52670 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SA...
449	32.24518...	192.168.0.147	192.168.0.115	TCP	66	52670 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1407804984
450	32.24552...	192.168.0.147	192.168.0.115	HTTP	407	GET /shell.php HTTP/1.1
451	32.24589...	192.168.0.115	192.168.0.147	TCP	66	80 → 52670 [ACK] Seq=1 Ack=342 Win=64896 Len=0 TSval=17019540...
452	32.24864...	192.168.0.115	192.168.0.147	TCP	74	53734 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 T...
453	32.24867...	192.168.0.147	192.168.0.115	TCP	74	80 → 53734 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SA...
454	32.24908...	192.168.0.115	192.168.0.147	TCP	66	53734 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1701954101
455	32.25470...	192.168.0.115	192.168.0.147	TCP	172	53734 → 80 [PSH, ACK] Seq=1 Ack=1 Win=64256 Len=106 TSval=170...
456	32.25472...	192.168.0.147	192.168.0.115	TCP	66	80 → 53734 [ACK] Seq=1 Ack=107 Win=65152 Len=0 TSval=14078049...
457	32.27156...	192.168.0.115	192.168.0.147	TCP	265	53734 → 80 [PSH, ACK] Seq=107 Ack=1 Win=64256 Len=199 TSval=1...
458	32.27159...	192.168.0.147	192.168.0.115	TCP	66	80 → 53734 [ACK] Seq=1 Ack=306 Win=65024 Len=0 TSval=14078050...
459	32.27581...	192.168.0.115	192.168.0.147	TCP	120	53734 → 80 [PSH, ACK] Seq=306 Ack=1 Win=64256 Len=54 TSval=17...
460	32.27585...	192.168.0.147	192.168.0.115	TCP	66	80 → 53734 [ACK] Seq=1 Ack=360 Win=65024 Len=0 TSval=14078050...
461	32.27781...	192.168.0.115	192.168.0.147	TCP	78	53734 → 80 [PSH, ACK] Seq=360 Ack=1 Win=64256 Len=12 TSval=17...
462	32.27786...	192.168.0.147	192.168.0.115	TCP	66	80 → 53734 [ACK] Seq=1 Ack=372 Win=65024 Len=0 TSval=14078050...
463	32.27812...	192.168.0.115	192.168.0.147	TCP	109	53734 → 80 [PSH, ACK] Seq=372 Ack=1 Win=64256 Len=43 TSval=17...
464	32.27813...	192.168.0.147	192.168.0.115	TCP	66	80 → 53734 [ACK] Seq=1 Ack=415 Win=65024 Len=0 TSval=14078050...
465	36.53758...	192.168.0.147	192.168.0.115	TCP	73	80 → 53734 [PSH, ACK] Seq=1 Ack=415 Win=65024 Len=7 TSval=140...
466	36.53792...	192.168.0.115	192.168.0.147	TCP	66	53734 → 80 [ACK] Seq=415 Ack=8 Win=64256 Len=0 TSval=17019583...
467	36.54057...	192.168.0.115	192.168.0.147	TCP	75	53734 → 80 [PSH, ACK] Seq=415 Ack=8 Win=64256 Len=9 TSval=170...

Transmission Control Protocol, Src Port: 52670, Dst Port: 80, Seq: 1, Ack: 1, Len: 341

Hypertext Transfer Protocol

- GET /shell.php HTTP/1.1\r\n
- Host: 192.168.0.115\r\n
- User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0\r\n
- Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8\r\n
- Accept-Language: en-US,en;q=0.5\r\n
- Accept-Encoding: gzip, deflate\r\n
- DNT: 1\r\n

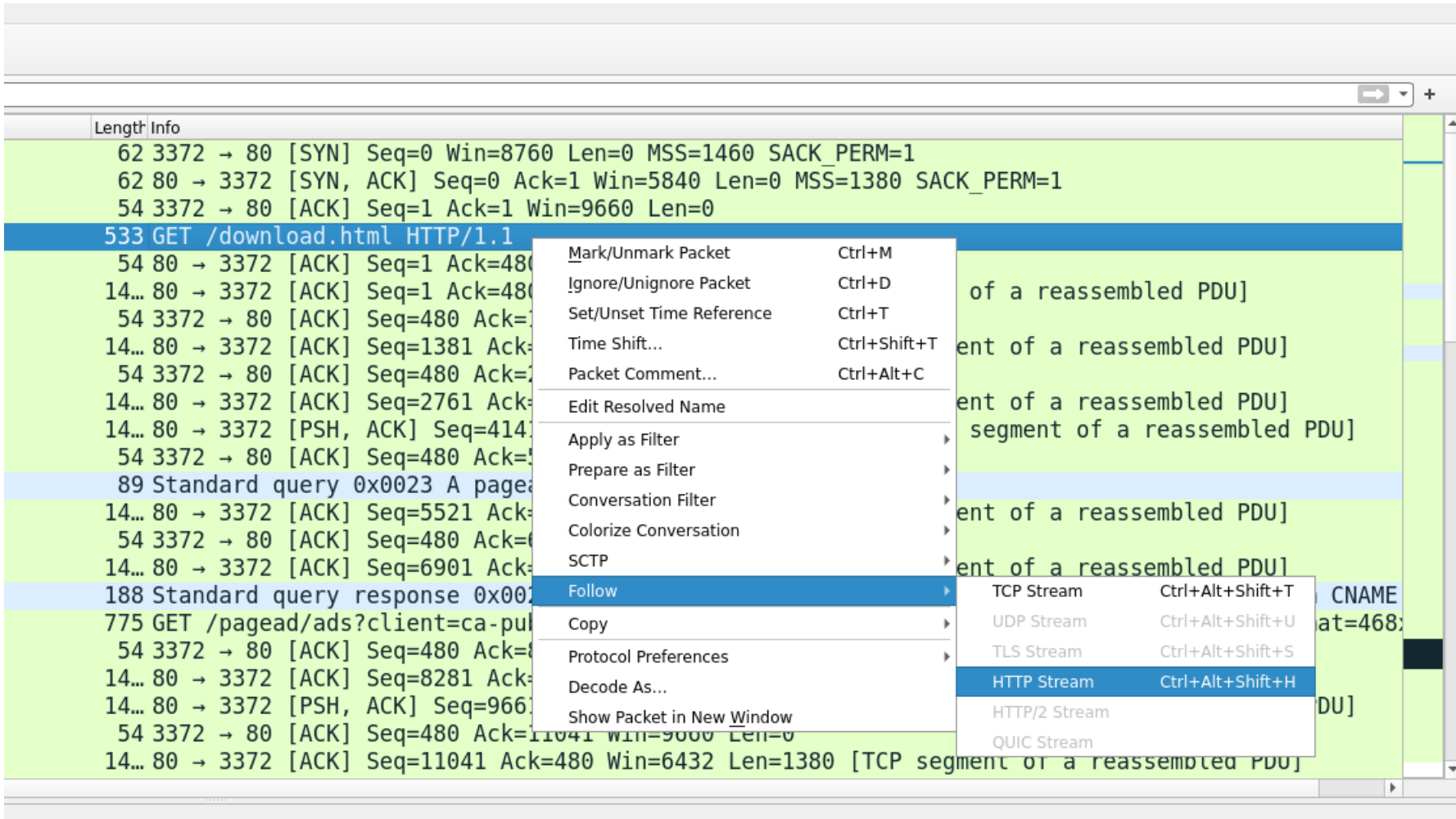
0000 08 00 27 92 a2 af 00 0c 29 4a b9 cd 08 00 45 00 ..'....)J....E.

0010 01 89 b0 1d 40 00 40 06 06 fb c0 a8 00 93 c0 a8@.@.....

0020 00 73 cd be 00 50 01 9f 1c bb 87 c6 14 06 80 18 .s...P.....

Capture.pcapng Packets: 907 · Displayed: 907 (100.0%) Profile: Default

II – Wireshark (Follow Stream)



The image shows the Wireshark network protocol analyzer interface. The packet list on the left contains several entries, including a GET request for /download.html. A right-click context menu is open over the selected packet, showing options like 'Follow', 'Copy', and 'Protocol Preferences'. The 'Follow' option is highlighted, and a sub-menu is visible showing 'HTTP Stream' as the selected option. The packet details pane on the right shows the structure of the selected packet, including the HTTP request line and headers.

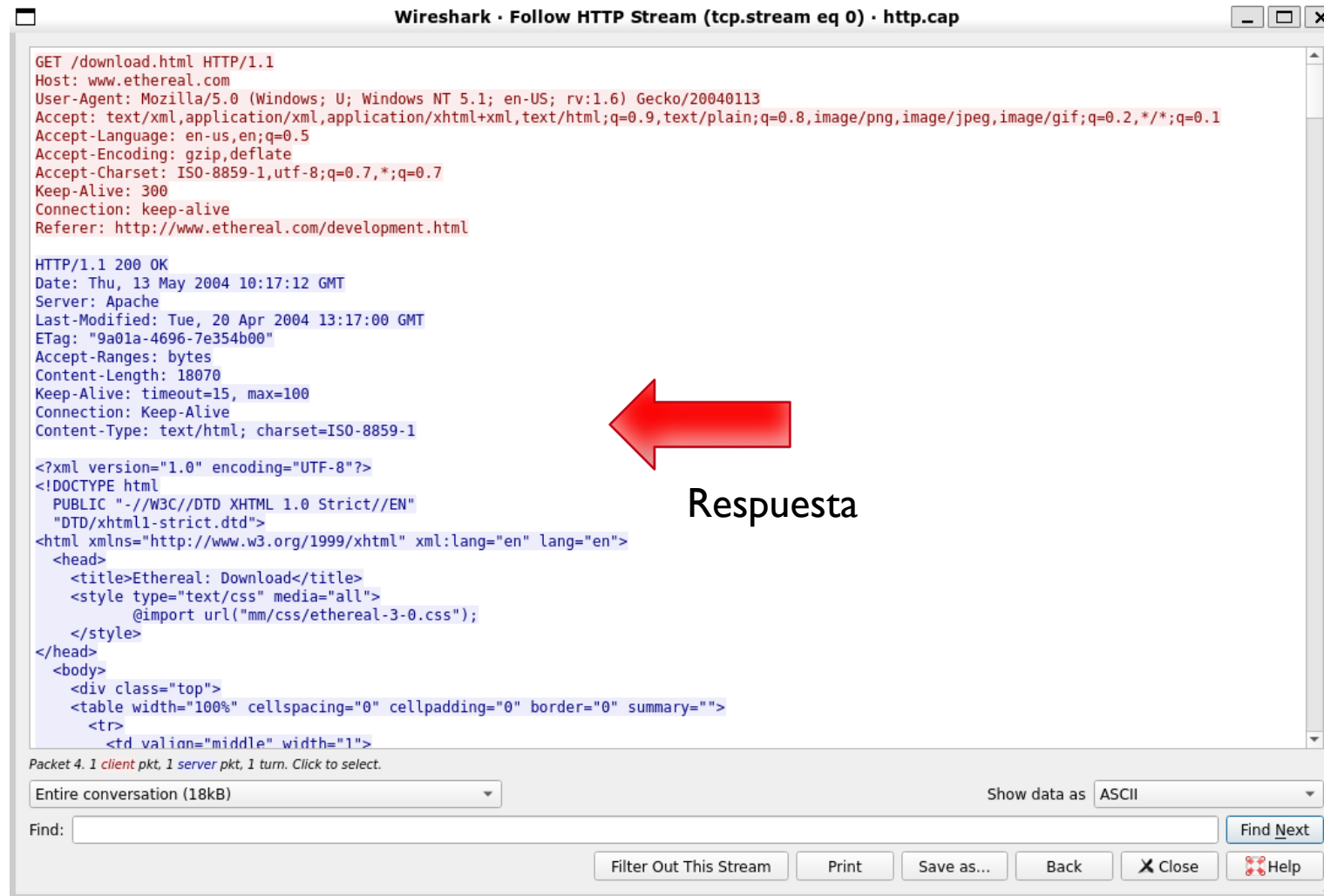
Length	Info
62	3372 → 80 [SYN] Seq=0 Win=8760 Len=0 MSS=1460 SACK_PERM=1
62	80 → 3372 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1380 SACK_PERM=1
54	3372 → 80 [ACK] Seq=1 Ack=1 Win=9660 Len=0
533	GET /download.html HTTP/1.1
54	80 → 3372 [ACK] Seq=1 Ack=480
14...	80 → 3372 [ACK] Seq=1 Ack=480
54	3372 → 80 [ACK] Seq=480 Ack=...
14...	80 → 3372 [ACK] Seq=1381 Ack=...
54	3372 → 80 [ACK] Seq=480 Ack=...
14...	80 → 3372 [ACK] Seq=2761 Ack=...
14...	80 → 3372 [PSH, ACK] Seq=414...
54	3372 → 80 [ACK] Seq=480 Ack=...
89	Standard query 0x0023 A pagea
14...	80 → 3372 [ACK] Seq=5521 Ack=...
54	3372 → 80 [ACK] Seq=480 Ack=...
14...	80 → 3372 [ACK] Seq=6901 Ack=...
188	Standard query response 0x002...
775	GET /pagead/ads?client=ca-pul
54	3372 → 80 [ACK] Seq=480 Ack=...
14...	80 → 3372 [ACK] Seq=8281 Ack=...
14...	80 → 3372 [PSH, ACK] Seq=966...
54	3372 → 80 [ACK] Seq=480 Ack=11041 Win=9660 Len=0
14...	80 → 3372 [ACK] Seq=11041 Ack=480 Win=6432 Len=1380 [TCP segment of a reassembled PDU]

- Mark/Unmark Packet Ctrl+M
- Ignore/Unignore Packet Ctrl+D
- Set/Unset Time Reference Ctrl+T
- Time Shift... Ctrl+Shift+T
- Packet Comment... Ctrl+Alt+C
- Edit Resolved Name
- Apply as Filter
- Prepare as Filter
- Conversation Filter
- Colorize Conversation
- SCTP
- Follow**
 - TCP Stream Ctrl+Alt+Shift+T
 - UDP Stream Ctrl+Alt+Shift+U
 - TLS Stream Ctrl+Alt+Shift+S
 - HTTP Stream Ctrl+Alt+Shift+H**
 - HTTP/2 Stream
 - QUIC Stream
- Copy
- Protocol Preferences
- Decode As...
- Show Packet in New Window

Opción muy útil para seguir la conversación HTTP

II – Wireshark (Follow Stream)

Petición



Wireshark · Follow HTTP Stream (tcp.stream eq 0) · http.cap

```
GET /download.html HTTP/1.1
Host: www.ethereal.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.6) Gecko/20040113
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,image/jpeg,image/gif;q=0.2,*/*;q=0.1
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: http://www.ethereal.com/development.html

HTTP/1.1 200 OK
Date: Thu, 13 May 2004 10:17:12 GMT
Server: Apache
Last-Modified: Tue, 20 Apr 2004 13:17:00 GMT
ETag: "9a01a-4696-7e354b00"
Accept-Ranges: bytes
Content-Length: 18070
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=ISO-8859-1

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
  <head>
    <title>Ethereal: Download</title>
    <style type="text/css" media="all">
      @import url("mm/css/ethereal-3-0.css");
    </style>
  </head>
  <body>
    <div class="top">
      <table width="100%" cellpadding="0" cellspacing="0" border="0" summary="">
        <tr>
          <td valign="middle" width="1">
```

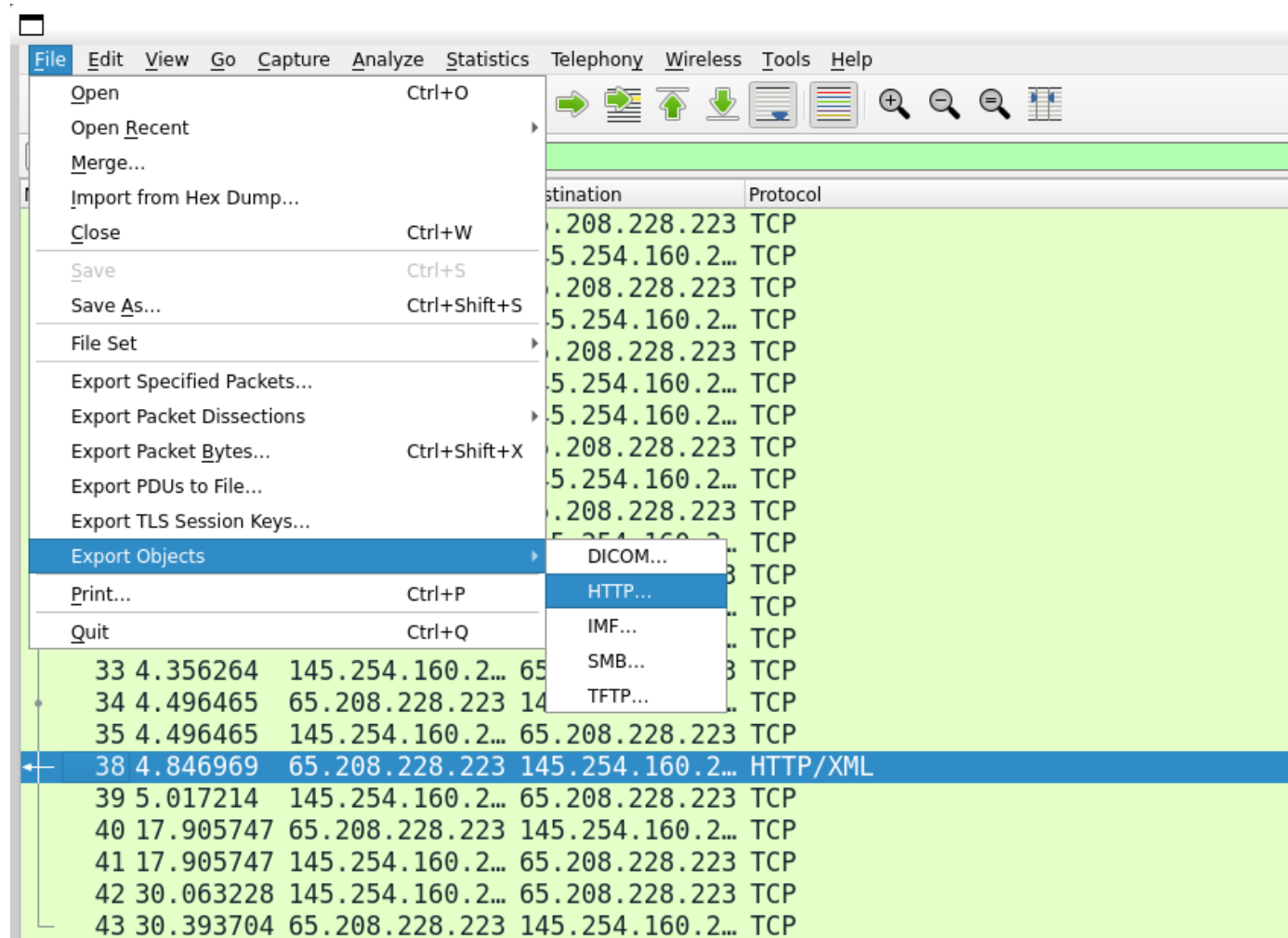
Packet 4. 1 client pkt, 1 server pkt, 1 turn. Click to select.

Entire conversation (18kB) Show data as ASCII

Find: Find Next

Filter Out This Stream Print Save as... Back Close Help

II – Wireshark (Export Objects)



Opción útil para exportar objetos de distintos protocolos

II – Wireshark (Export Objects)

Wireshark · Export · HTTP object list

Packet	Hostname	Content Type	Size	Filename
54	www.msftncsi.com	text/plain	14 bytes	ncsi.txt
132	api.bing.com	text/html	1,305 bytes	qsml.aspx?que
163	api.bing.com	text/html	1,346 bytes	qsml.aspx?que
177	api.bing.com	text/html	1,369 bytes	qsml.aspx?que
198	api.bing.com	text/html	1,398 bytes	qsml.aspx?que
212	google.com	text/html	219 bytes	/
226	www.google.com	text/html	231 bytes	/
1858	www.google.com	text/html	1,058 bytes	url?sa=t&rct=
1904	www.blupproducts.com	text/html	19 kB	/
1955	www.blupproducts.com	text/css	7,321 bytes	default_iceme
1972	www.blupproducts.com	text/css	331 bytes	default_notjs.c
2109	www.blupproducts.com	text/css	63 kB	widgetkit-2410
2136	www.blupproducts.com	application/x-javascript	4,707 bytes	core-816de4c1
2139	www.blupproducts.com	application/x-javascript	657 bytes	caption-5e0b3
2280	www.blupproducts.com	application/x-javascript	20 kB	widgetkit-34c2
2390	www.blupproducts.com	application/x-javascript	18 kB	cufon-yui-1d10
2545	www.blupproducts.com	application/x-javascript	95 kB	mootools-core
2560	www.blupproducts.com	application/x-javascript	93 kB	jquery-7ae67c
2689	www.blupproducts.com	application/x-javascript	4,784 bytes	core.js
2728	platform.linkedin.com	text/javascript	3,768 bytes	in.js
2743	www.blupproducts.com	text/css	132 kB	template-897f
2784	www.blupproducts.com	application/x-javascript	22 kB	template-3f20
2898	www.blupproducts.com	image/png	19 kB	facebook.png
2990	www.blupproducts.com	image/png	22 kB	Twitter.png
3060	www.blupproducts.com	image/png	44 kB	googleplus.pn
3066	s.amazon-adsystem.com	image/gif	43 bytes	iui3?d=3p-hbc
3145	www.blupproducts.com	image/png	19 kB	mail.png

Text Filter:

II – Wireshark (Filtros)

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 lqaz2wsx
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 (Retos)



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

Antes de analizar tenemos que crear un caso

I. Nombre del caso

II. Descripción

III. Participantes en la investigación

CREATE A NEW CASE

1. **Case Name:** The name of this investigation. It can contain only letters, numbers, and symbols.

2. **Description:** An optional, one line description of this case.

3. **Investigator Names:** The optional names (with no spaces) of the investigators for this case.

a.	<input type="text"/>	b.	<input type="text"/>
c.	<input type="text"/>	d.	<input type="text"/>
e.	<input type="text"/>	f.	<input type="text"/>
g.	<input type="text"/>	h.	<input type="text"/>
i.	<input type="text"/>	j.	<input type="text"/>

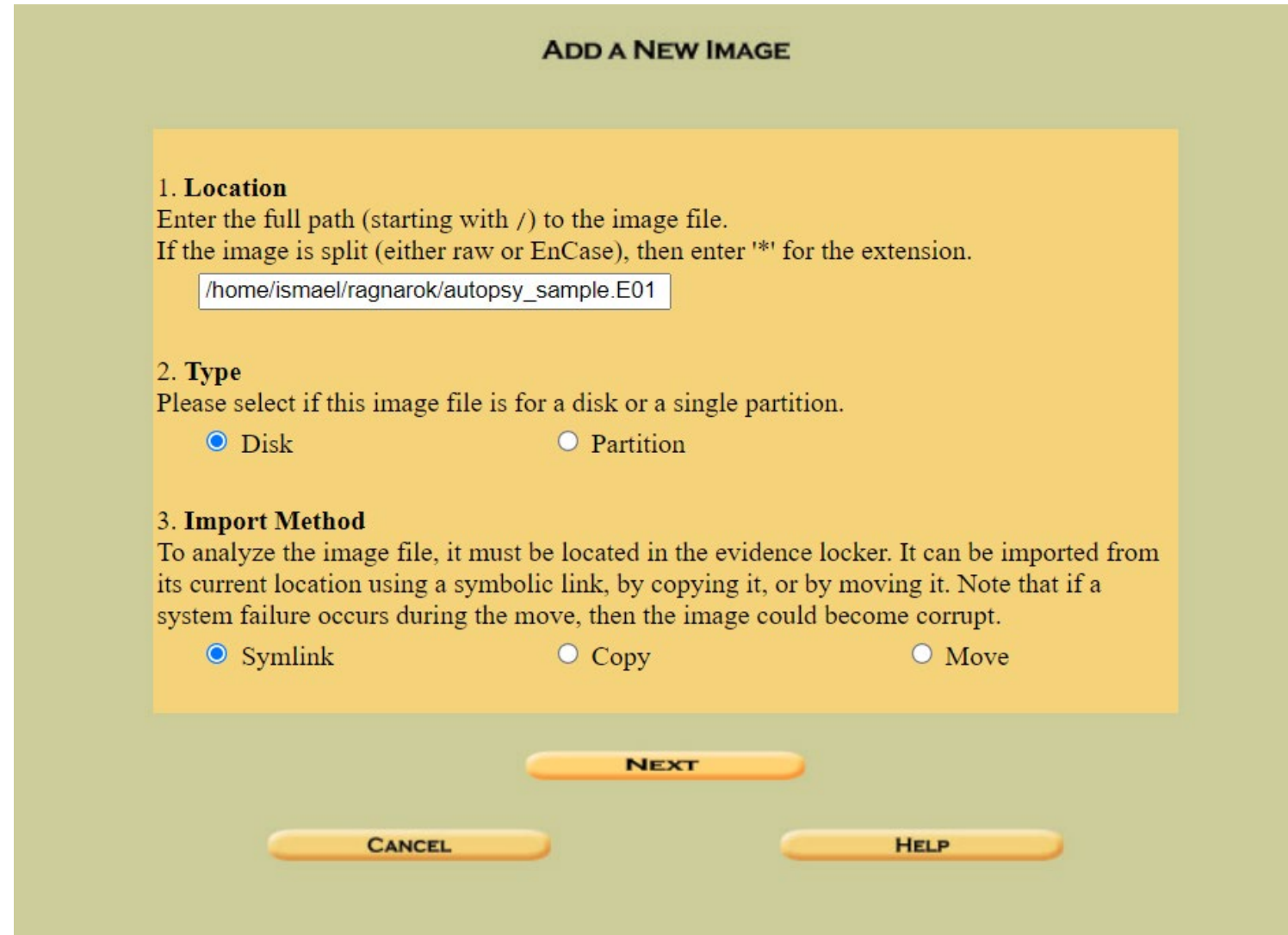
NEW CASE **CANCEL** **HELP**

III - Autopsy (Añadir nueva imagen)

Para agregar una imagen al caso simplemente escribimos la **ruta completa al fichero a analizar**.

Después, indicar si el fichero es una **imagen de disco entera** o una **partición** (si no estamos seguros lo dejamos en disco)

De método de importación, elegir el que más convenga (por temas de espacio elegí enlace simbólico)



The screenshot shows the 'ADD A NEW IMAGE' dialog box in the Autopsy software. The dialog has a yellow background and is titled 'ADD A NEW IMAGE'. It contains three sections: 1. Location, 2. Type, and 3. Import Method. In the Location section, the full path '/home/ismael/ragnarok/autopsy_sample.E01' is entered in a text box. In the Type section, the 'Disk' radio button is selected. In the Import Method section, the 'Symlink' radio button is selected. At the bottom of the dialog, there are three buttons: 'NEXT', 'CANCEL', and 'HELP'.

ADD A NEW IMAGE

1. Location
Enter the full path (starting with /) to the image file.
If the image is split (either raw or EnCase), then enter '*' for the extension.

2. Type
Please select if this image file is for a disk or a single partition.
☒ Disk ☐ Partition

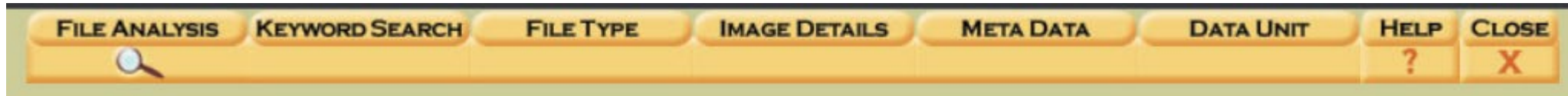
3. Import Method
To analyze the image file, it must be located in the evidence locker. It can be imported from its current location using a symbolic link, by copying it, or by moving it. Note that if a system failure occurs during the move, then the image could become corrupt.
☒ Symlink ☐ Copy ☐ Move

NEXT **CANCEL** **HELP**

III – Autopsy (Analizar disco)



III – Autopsy (Tipos de análisis)



Tenemos varios tipos de análisis:

I. File Analysis -> Análisis del sistema de ficheros

II. Keyword Search -> Un "strings" a lo bestia

III. FileType -> "file" a lo bestia, intenta detectar ficheros con extensión cambiada

IV. MetaData -> Útil para recuperar

V. Data Unit -> Te permite ver datos de distintas formas, como hexdump, dd, etc...

III – Autopsy (File Analysis)

Directory Seek Enter the name of a directory that you want to view. C:/	✓	r / r	_54402.EXE	2009-11-20 10:31:36 (CET)	2009-11-20 00:00:00 (CET)	2009-11-20 10:49:30 (CET)
	✓	d / d	_604468_/	2009-11-20 10:51:54 (CET)	2009-11-20 00:00:00 (CET)	2009-11-20 10:31:34 (CET)
VIEW		d / d	Log/	2009-12-07 08:05:22 (CET)	2009-12-07 00:00:00 (CET)	2009-12-07 08:05:20 (CET)
File Name Search Enter a Perl regular expression for the file names you want to find.		r / r	M57biz.jpg	2009-11-17 08:50:26 (CET)	2009-12-07 00:00:00 (CET)	2009-11-17 08:50:25 (CET)
		r / r	patentauto.py	2009-11-17 13:37:00 (CET)	2009-11-17 00:00:00 (CET)	2009-11-16 14:16:49 (CET)
SEARCH		r / r	patentterms.txt	2009-11-16 14:29:38 (CET)	2009-11-24 00:00:00 (CET)	2009-11-14 17:43:57 (CET)
ALL DELETED FILES		r / r	R54402.EXE	2009-11-20 10:31:44 (CET)	2009-12-07 00:00:00 (CET)	2009-11-20 10:31:34 (CET)
EXPAND DIRECTORIES		r / r	TERRYS WORK (Volume Label Entry)	2009-11-17 13:47:24 (CET)	0000-00-00 00:00:00 (UTC)	0000-00-00 00:00:00 (UTC)
		r / r	urlscopyright.txt	2009-11-17 10:40:56 (CET)	2009-11-24 00:00:00 (CET)	2009-11-17 10:40:57 (CET)
		r / r	urlscryptography.txt	2009-11-16 10:22:50 (CET)	2009-11-24 00:00:00 (CET)	2009-11-16 10:22:51 (CET)
		r / r	urlspatents.txt	2009-11-17 10:40:56 (CET)	2009-11-24 00:00:00 (CET)	2009-11-17 10:40:57 (CET)
		r / r	urlspersona.txt	2009-11-14 17:43:14 (CET)	2009-11-24 00:00:00 (CET)	2009-11-14 17:41:55 (CET)
		r / r	urlstime_machine.txt	2009-11-16 10:22:50 (CET)	2009-11-24 00:00:00 (CET)	2009-11-16 10:22:51 (CET)
		r / r	vnc-4_1_3-x86_win32.exe	2008-10-15 17:14:08 (CEST)	2009-12-07 00:00:00 (CET)	2008-10-15 17:14:08 (CEST)
		r / r	webauto.py	2009-11-16 14:23:38 (CET)	2009-11-24 00:00:00 (CET)	2009-11-14 17:39:19 (CET)
	✓	r / r	xpadvancedkeylogger.exe	2009-12-03 09:40:44 (CET)	2009-12-07 00:00:00 (CET)	2009-12-03 09:41:16 (CET)

III – Autopsy (File Analysis)

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

ASCII (display - report) * Hex (display - report) * ASCII Strings (display - report) * Export * Add Note

File Type: Python script, ASCII text executable

Contents Of File: C:/patentauto.py

```
#!/usr/bin/python

__author__="LCDR Kris Kearton"
__date__="$Aug 24, 2009 7:42:41 PM$"
# class: CS4920 ADOEX
# System info: Running on OS 10.6 python ver 2.6.2
# Setup information:
# (1) Install MozRepl Plugin at:
# http://wiki.github.com/bard/mozrepl
# Once installed, ensure in Firefox under tools MozRepl is started
#
# Summary: MozRepl needs to telnet to the browser via port 4242. Once connected the port can program
# can issue commands directly to the web browser. This program gets the list of urls from the text file.
# Then randomly picks a URL and surfs it for background noise.

import time
import csv
import telnetlib
import robotparser
import os
import random

#
#connect to MozRepl and fetch HTML
#
def connect_mozrepl(url_addr):
    quit = False
    t = telnetlib.Telnet("localhost", 4242)
    t.read_until("repl>")

    #verifies page was accepted
    rp = robotparser.RobotFileParser()
    fetched = rp.can_fetch("*", url_addr)
    print fetched
    state = True
    while(state==True):
        if fetched==True:
```

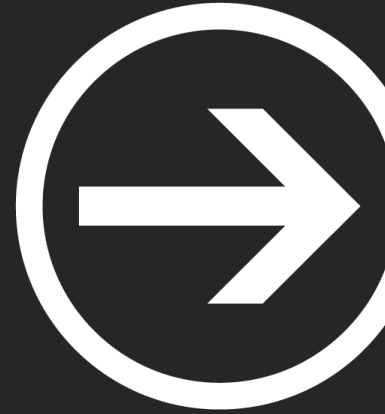
III – Autopsy (File Analysis)

Current Directory: [C:/](#) /Log/

[ADD NOTE](#)
[GENERATE MD5 LIST OF FILES](#)

DEL	Type	NAME	WRITTEN	ACCESSED	CREATED	SIZE	UID	GID	META
	dir / in								
	d / d	../	0000-00-00 00:00:00 (UTC)	0000-00-00 00:00:00 (UTC)	0000-00-00 00:00:00 (UTC)	4096	0	0	2
	d / d	../	2009-12-07 08:05:22 (CET)	2009-12-07 00:00:00 (CET)	2009-12-07 08:05:20 (CET)	643072	0	0	72
	r / r	2009-12-03.htm	2009-12-03 23:59:10 (CET)	2009-12-07 00:00:00 (CET)	2009-12-07 08:05:20 (CET)	441396	0	0	4231
	r / r	2009-12-03_00036d9f_big.jpg	2009-12-03 19:11:10 (CET)	2009-12-07 00:00:00 (CET)	2009-12-07 08:05:21 (CET)	74965	0	0	4235
	r / r	2009-12-03_00036d9f_small.jpg	2009-12-03 19:11:10 (CET)	2009-12-07 00:00:00 (CET)	2009-12-07 08:05:21 (CET)	4369	0	0	4239
	r / r	2009-12-03_0005425f_big.jpg	2009-12-03 19:13:10 (CET)	2009-12-07 00:00:00 (CET)	2009-12-07 08:05:21 (CET)	74958	0	0	4243
	r / r	2009-12-03_0005425f_small.jpg	2009-12-03 19:13:10 (CET)	2009-12-07 00:00:00 (CET)	2009-12-07 08:05:21 (CET)	4369	0	0	4247
	r / r	2009-12-03_0007171f_big.jpg	2009-12-03 19:15:10 (CET)	2009-12-07 00:00:00 (CET)	2009-12-07 08:05:21 (CET)	74971	0	0	4251
	r / r	2009-12-03_0007171f_small.jpg	2009-12-03 19:15:10 (CET)	2009-12-07 00:00:00 (CET)	2009-12-07 08:05:21 (CET)	4369	0	0	4255
	r / r	2009-12-03_0008ebdf_big.jpg	2009-12-03 19:17:10 (CET)	2009-12-07 00:00:00 (CET)	2009-12-07 08:05:22 (CET)	74957	0	0	4259
	r / r	2009-12-03_0008ebdf_small.jpg	2009-12-03 19:17:10 (CET)	2009-12-07 00:00:00 (CET)	2009-12-07 08:05:22 (CET)	4369	0	0	4263
	r / r	2009-12-03_000ac09f_big.jpg	2009-12-03 19:19:10 (CET)	2009-12-07 00:00:00 (CET)	2009-12-07 08:05:22 (CET)	74965	0	0	4267
	r / r	2009-12-03_000ac09f_small.jpg	2009-12-03 19:19:10 (CET)	2009-12-07 00:00:00 (CET)	2009-12-07 08:05:22 (CET)	4369	0	0	4271
	r / r	2009-12-03_000c955f_big.jpg	2009-12-03 19:21:10 (CET)	2009-12-07 00:00:00 (CET)	2009-12-07 08:05:22 (CET)	12663	0	0	4275
	r / r	2009-12-03_000c955f_small.jpg	2009-12-03 19:21:10 (CET)	2009-12-07 00:00:00 (CET)	2009-12-07 08:05:22 (CET)	1186	0	0	4279
	r / r	2009-12-03_000e6a1f_big.jpg	2009-12-03 19:23:10 (CET)	2009-12-07 00:00:00 (CET)	2009-12-07 08:05:22 (CET)	12538	0	0	4283
	r / r	2009-12-03_000e6a1f_small.jpg	2009-12-03 19:23:10 (CET)	2009-12-07 00:00:00 (CET)	2009-12-07 08:05:23 (CET)	1165	0	0	4287

Navegación por distintos directorios del disco



Modulo II: Forense

Ismael Gómez Esquilichi y Alejandro Bermejo Pérez



Universidad
Rey Juan Carlos