[Windows] - Wardriving Field Manual

Manual técnico para participar del wardriving sin mucho esfuerzo.

Lo primero que debemos entender es que el wardriving no se trata de capturar el tráfico de las redes inalámbricas, si no de captar su presencia en el medio en conjunto con su ubicación geográfica.

Para esto es necesario contar con el hardware necesario para poder procesar las señales y obtener coordenadas GPS.

Durante el desarrollo de este tutorial iremos preparando paso a paso el entorno para wardriving.

0 - Requisitos:

Mínimos:

1. CPU: i5 de 4ta generación o equivalente.
2. RAM: 4gb DDR3.
3. NIC: Cualquiera compatible con modo monitor.

Recomendados:

1. CPU: i5 de 7ma Generación o superior.
2. RAM: 8gb DDR3 o superior.
3. NIC: Cualquiera compatible con modo monitor, doble banda (2.4 - 5.8), MIMO y 2 conectores RP-SMA para recambio de antenas.

Para sistemas embebidos:

Cualquiera parecido a la Raspberry Pi 3b+ o Raspberry Pi Zero W 2 funcionará.
El requisito mínimo indispensable es de 2gb de RAM si van a utilizar antenas de alta ganancia, dado que el buffer se llena demasiado rápido por la cantidad de redes que debe procesar, lo cual causará que Kismet crashee y deje de procesar, afectando el número de redes totales al final del wardriving que uno pueda capturar.
Hablaremos de este problema más adelante.

Software:

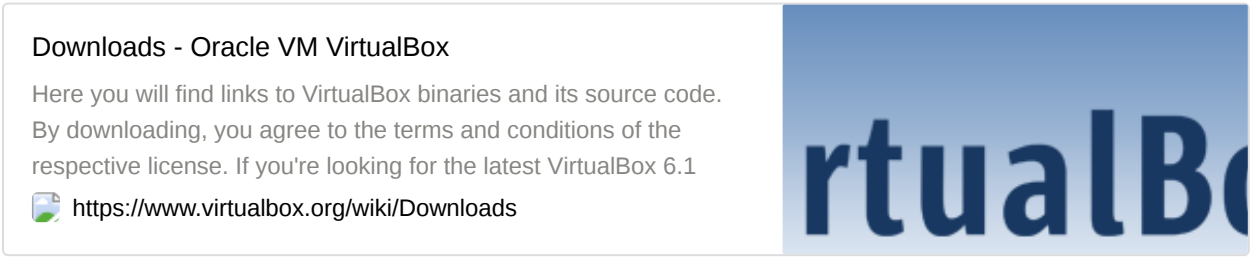
1. Sist. Operativo: Recomendamos ampliamente Kali Linux.
2. Airmon-ng (incl. en aircrack-ng).
3. gpsd y gpsd-clients.
4. netcat (nc).
5. Kismet.
6. ADB (Android Debug Bridge)
7. Módulos de Kernel y Drivers USB para conectar los periféricos que poseemos. Algunos no son plug and play.
8. ShareGPS (Android).
9. WiGLE (Android - iOS).
10. Software de virtualización en caso de no correr la maquina nativamente (o live).
    a. MacOS: VMWare Fusion Personal License (desaconsejamos VirtualBox). No se ha testeado en Apple Silicon, solo Intel.
    b. Windows / Linux: Recomendamos VirtualBox.

1 - Instalando Virtual Box en Windows:

Instalar Oracle Virtualbox dentro de un sistema operativo Windows es una tarea bastante sencilla.
Lo primero que debemos hacer es asegurarnos de que la computadora cuenta con la virtualización de 64 bits encendida.

En caso de no tenerlo, debemos entrar al BIOS y activarlo (varía en cada fabricante y modelo).

Una vez que chequeamos esto, debemos proceder a bajar virtualbox desde el siguiente link:



Una vez descargado e instalado, debemos agregar el VirtualBox Extension Pack, que es el que nos permitirá conectividad con diversos dispositivos USB desde la máquina virtual.
Caso contrario no podremos utilizar las NIC, ni ningún dispositivo que no sea USB 1.0.

Para descargarlas, debemos ir al link anterior y dirigirnos a la sección “VirtualBox 7.X.X Oracle VM VirtualBox Extension Pack”. Para instalarlas debemos hacer doble click sobre ellas y se agregarán automáticamente a VirtualBox (se abrirá el programa).
Una vez finalizado esto, virtualbox está listo para ser utilizado como hipervisor.

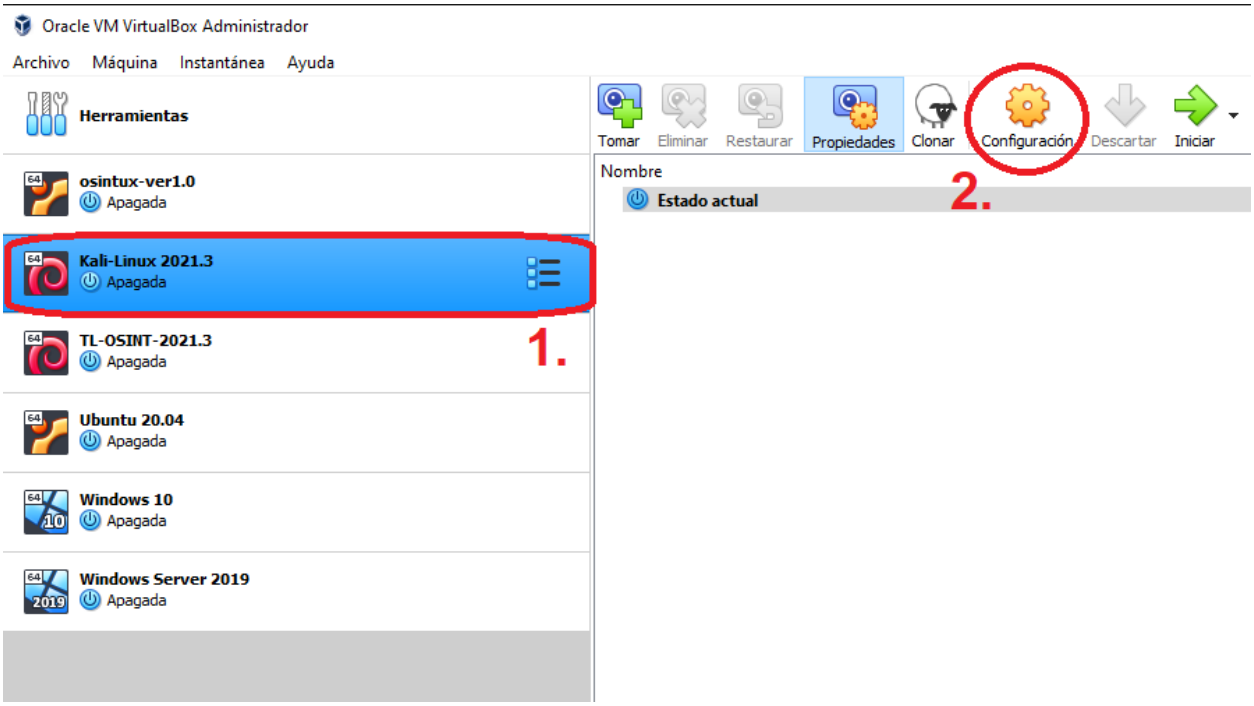
2 - Configurando el entorno para wardriving:

Debemos realizar una serie de configuraciones previas antes de arrancar que abarcan la instalación del sistema operativo y los distintos paquetes que utilizaremos para nuestro fin.
Comencemos.

Configuración de entorno en VirtualBox

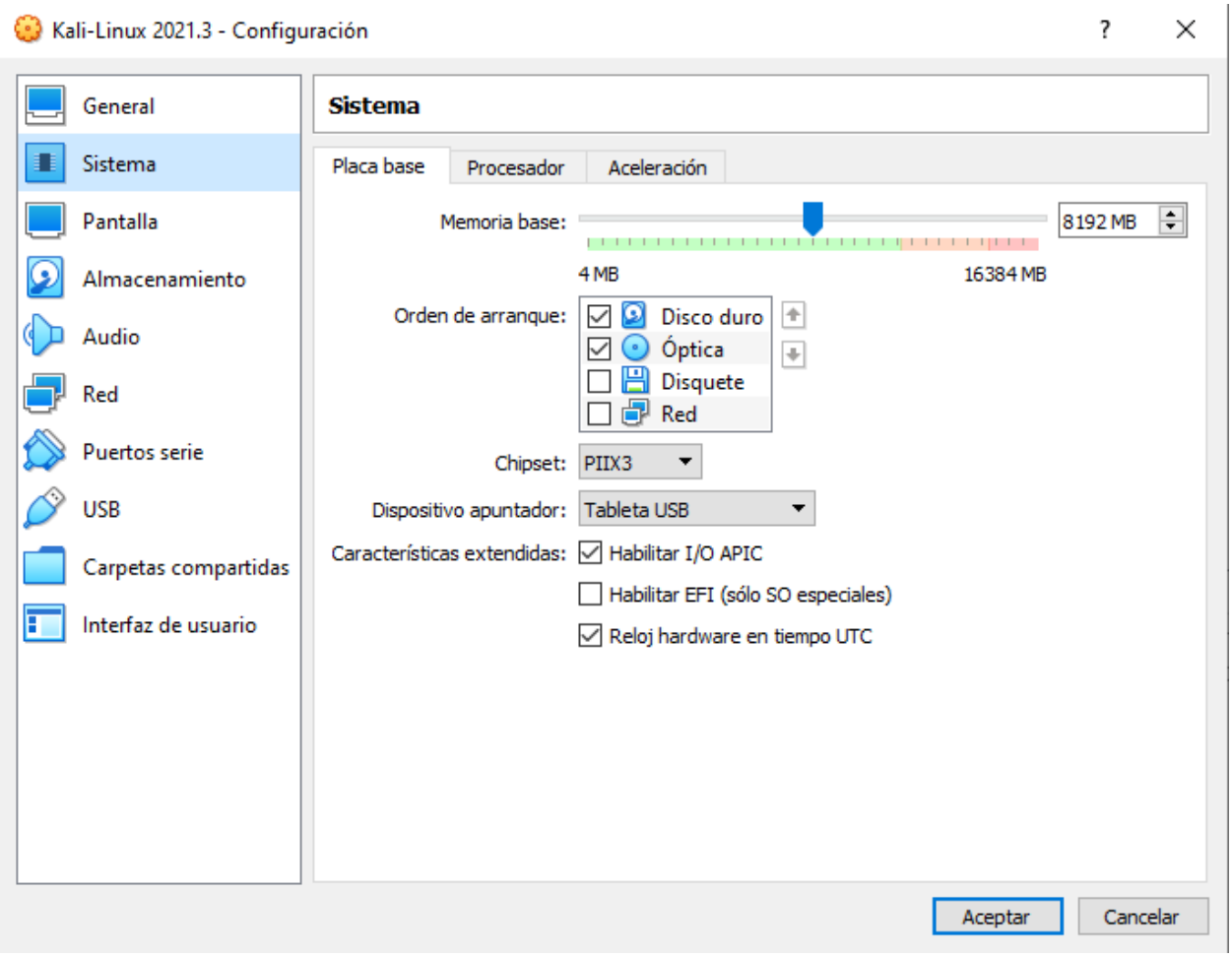
Para los usuarios que tienen VirtualBox, nos van a figurar dos archivos: uno con extensión VDI y otro con extensión VBOX.
Si ejecutamos el archivo .vbox, se abrirá automáticamente virtualbox y se cargará el archivo VDI (Virtual Disk Image, es decir el disco duro virtual de KaliLinux) y tendremos nuestra máquina casi lista.
Antes de ejecutarla, debemos modificar los recursos de hardware de la misma para poder utilizarla.

Para realizar esto, debemos seleccionar la máquina que queremos modificar, luego en configuración como se muestra a continuación.

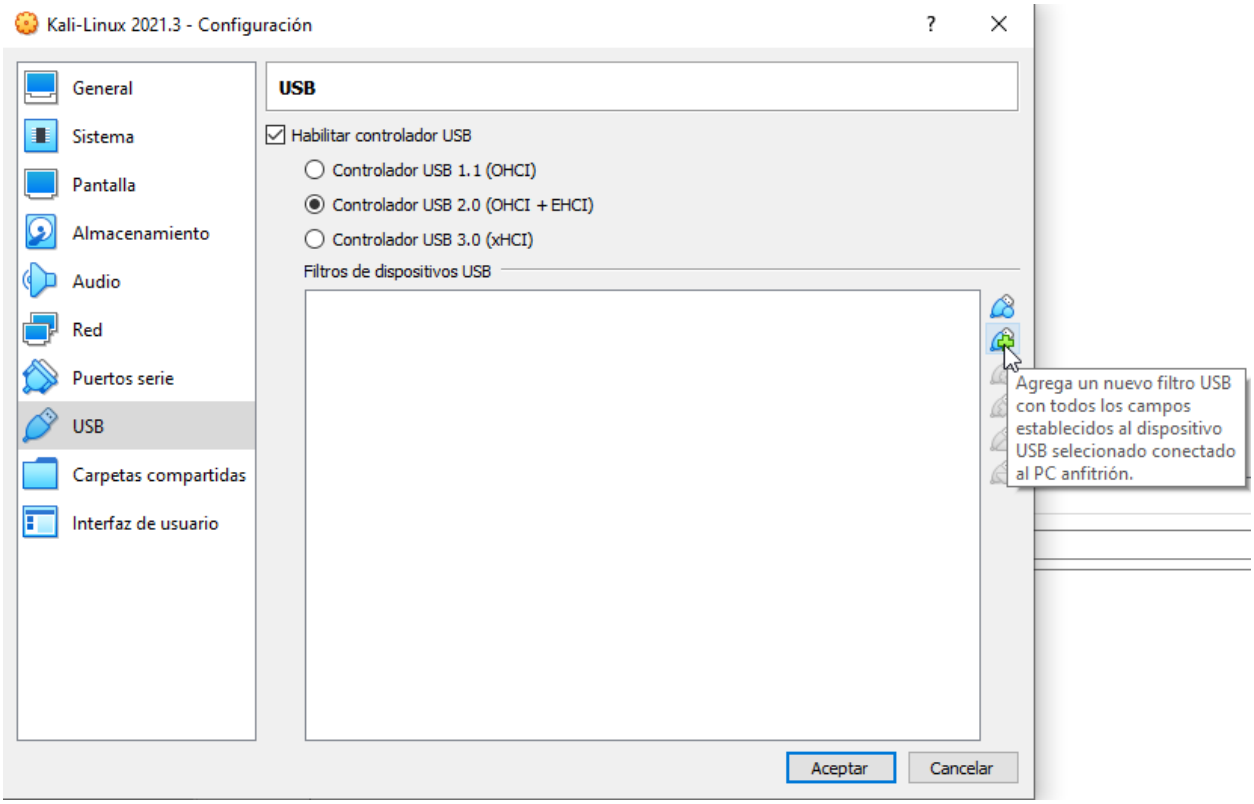


1- Seleccionamos la VM.
2- Clickeamos en configuración.

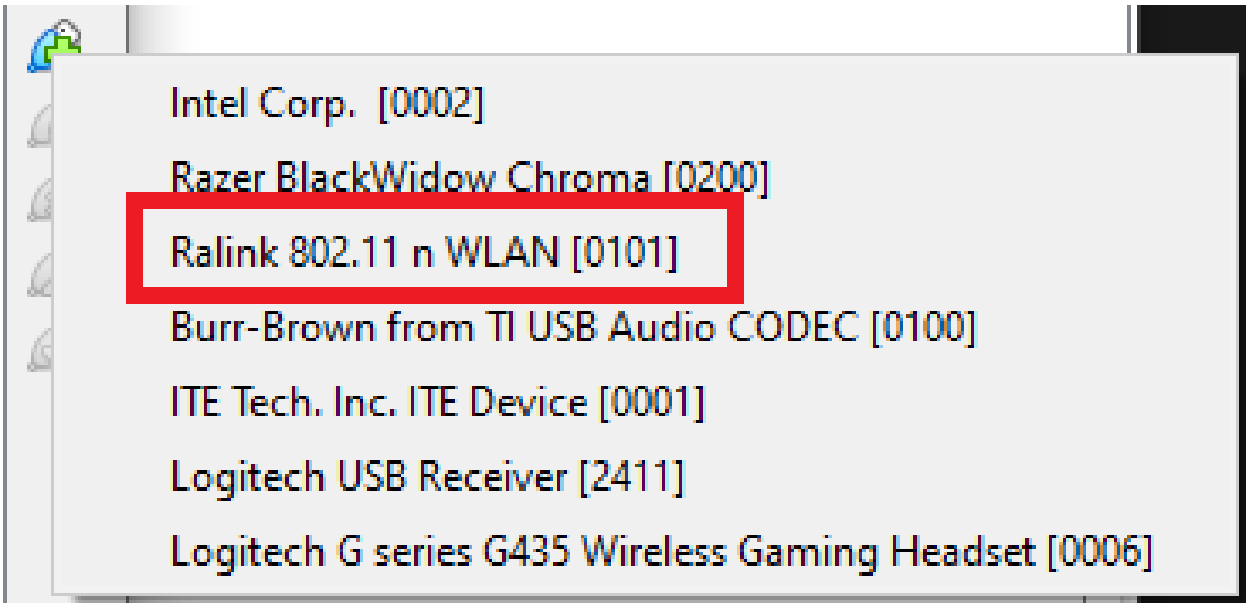
Se nos abrirá el siguiente menú, donde debemos dirigirnos a la sección “Sistema”, dónde podremos aumentar el RAM y vCPU asignados a la máquina virtual en cuestión. Se recomienda asignar la mitad de los recursos de nuestra computadora para un optimo uso.
Para asignar más núcleos de procesador, nos dirigimos a la solapa “Procesador”.



Una vez hecho esto, procedemos a dar acceso a los dispositivos USB que vayamos a utilizar para la actividad.
Esto lo realizamos mediante la solapa “USB” en este mismo menú:
Para agregar el dispositivo USB en cuestión, debemos clickear en el ícono que señala el mouse en la siguiente imagen. El mismo abrirá un pequeño menú que listará todos los dispositivos USB conectados a nuestra computadora.
Generalmente no nos aparecerá la marca y modelo de nuestra antena, si no el chipset de la misma.
En este caso se puede observar que la placa que se está agregando es una Ralink.
Por último, es recomendable que utilicemos el controlador USB 2.0 o superior (esto afectará a todos los dispositivos USB de la máquina virtual).

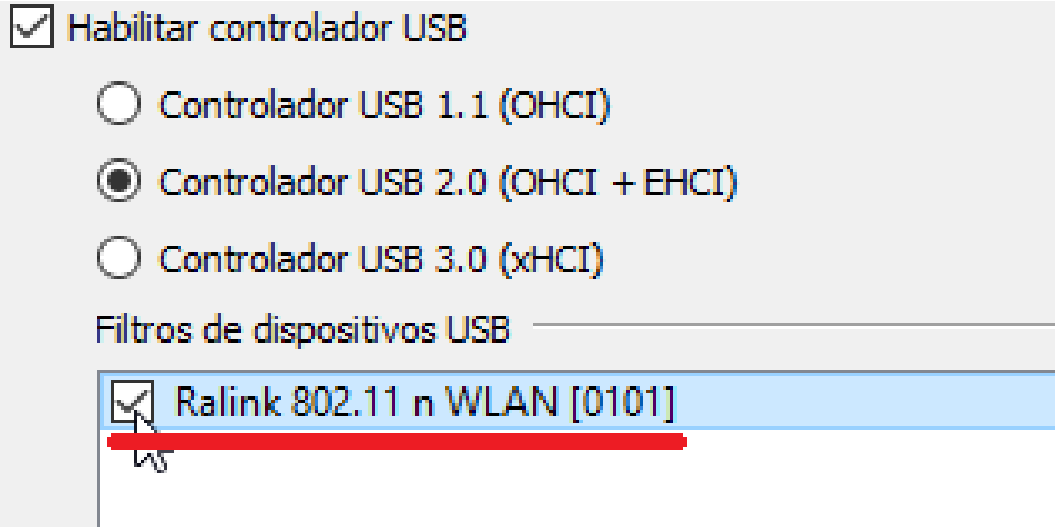


Clickeamos en el icono de agregar un filtro USB. Esto realiza la conexión automáticamente, pero en caso de que necesitemos configurar otros parámetros, se puede agregar el filtro manualmente (solo para usuarios avanzados).



Nuestro NIC cuenta con un chipset Ralink 3060, el cual observamos aquí.

Una vez añadido, lo veremos en la lista de dispositivos que antes estaba vacía.
Si no tiene una tilde marcada a la izquierda en su checkbox, debemos marcarlo para que se conecte a nuestra VM, caso contrario no se conectará.
Debemos repetir el mismo proceso para todos los dispositivos que deseamos conectar mediante USB.



Aquí observamos que el checkbox tiene su tilde y está listo para ser utilizado.

Con esto ya habremos finalizado la configuración del entorno VirtualBox.
Ahora nos queda lanzar la máquina virtual y comenzar a configurarla.

3 - Configurando Kali Linux para Wardriving:

Instalación de paquetes necesarios:

Una vez que hayamos lanzado nuestra máquina virtual con todos los periféricos necesarios conectados, es hora de comenzar a instalar los paquetes necesarios para el wardriving.

Lo primero que debemos hacer es ejecutar `sudo apt update` a fin de actualizar los repositorios.
No ejecuten "upgrade" ni similar fuera de sus casas debido a que la conexión probablemente no sea óptima en el centro de convenciones y demorará muchísimo en actualizar el sistema entero.

Lo segundo que debemos realizar es la instalación de varios programas, los cuales se encuentran todos en los repositorios de Kali.
Para ello ejecutamos:

```
sudo apt install kismet gpsd gpsd-clients adb aircrack-ng -y
```

Estos son los paquetes que vamos a necesitar para las diversas tareas que vamos a realizar previo al wardriving.

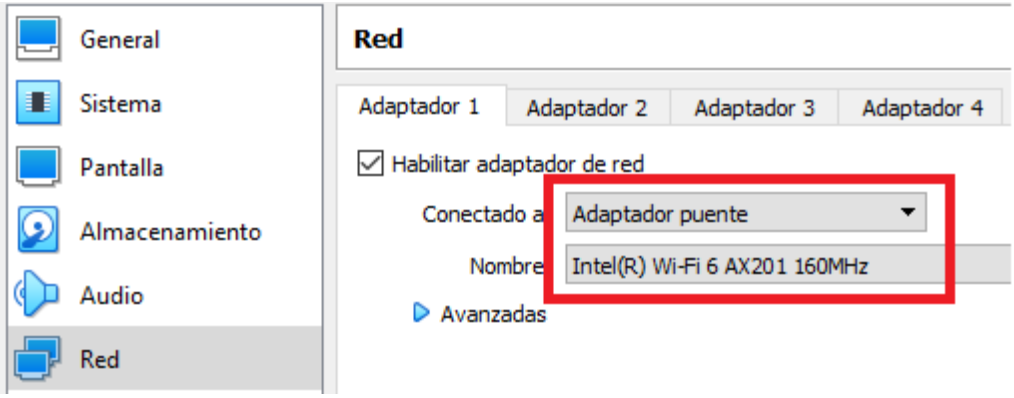
4 - Configurando el GPS:

Una vez que ya tenemos instalados los paquetes, tenemos tres maneras de recibir las coordenadas GPS:

- Mediante un servidor en una red LAN creada exclusivamente para el consumo del GPS.
- Mediante un adaptador GPS USB como el GMouse.
- Mediante una app llamada “ShareGPS” en Android y utilizando USB Tethering.

GPS Mediante Red LAN:

Para consumir el GPS del servidor que estamos poniendo a su disposición, debemos configurar Kali Linux previamente para que utilice el adaptador de red en modo puente. Esto nos brindará visibilidad directa de la red a la que se conecta nuestra computadora física (SO Anfitrión) para poder consumir el GPS.
Para esto debemos apagar nuestra máquina virtual, volver al menu de configuración e ir a la sección de red y seleccionar el adaptador de red built-in de nuestro equipo y pasarlo a modo “Bridge” como se muestra en la imagen a continuación.
Es importante que el adaptador que estemos utilizando sea el embebido en nuestra maquina y no el NIC que utilizaremos para kismet, caso contrario nos quedaremos sin capacidad de captura.



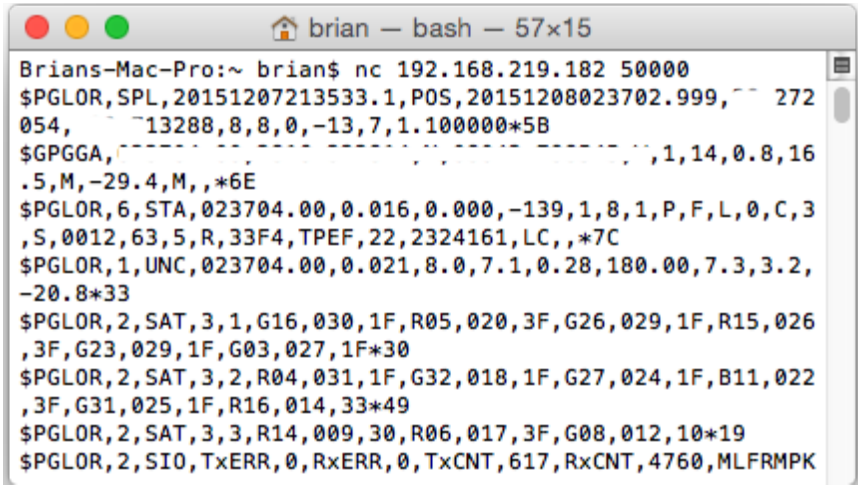
Si lo tienen configurado en inglés, debería decir algo como “Bridge Adaptor”.

Ahora volvemos a lanzar Kali Linux y verificamos que nuestra red sea la que se asignó por el Hotspot WiFi provisto por los coordinadores.

Una vez dentro de Kali Linux, vamos a testear que recibimos correctamente el flujo NMEA del GPS con Netcat. Para ello ejecutamos lo siguiente (La IP indicada puede variar según como configuremos el hotspot los coordinadores):

```
nc 10.3.141.202 50000
```

La IP indicada puede variar según como configuremos el hotspot los coordinadores.
Y debemos recibir un output parecido al siguiente:



Si todo está OK y el output se nos muestra como indica la imagen de arriba, procedemos a testear con GPSD.
Para ello ejecutaremos los siguientes comandos:

```
sudo killall gpsd
gpsd -D5 -N -n -b tcp://10.3.141.202:50000
```

Debemos tener un output parecido al de la imagen anterior.

Ahora que sabemos que el GPSD está recibiendo el stream del servidor GPS, podemos proceder a bajarlo de nuevo con el comando “killall gpsd” para relanzarlo en modo background.
Para ello debemos ejecutar:

`gpsd -b -n tcp://10.3.141.202:50000`

El puerto por defecto del GPSD será el 2794, esto lo podemos modificar con el flag “-S xxxx” donde XXXX es el puerto que deseemos utilizar.

GPS Mediante USB:

Para este método debemos contar con el paquete ADB (Android Debug Bridge) un celular Android y el cable USB del mismo.
Verificar que primero tengamos instalados los drivers USB de nuestro celular a fin de que funcione correctamente o con todas sus funcionalidades.

Antes de conectar el celular, debemos agregarlo a la interface USB como hicimos con la placa de red en pasos anteriores. Recordemos que para ello la maquina virtual debe estar apagada.
Una vez que esto está hecho, volvemos a prender la máquina virtual y nuestro celular estará directamente conectado a ella.

Ahora iremos al PlayStore y descargaremos la app “ShareGPS” la cual nos permitirá conectarnos al GPS del celular y servirlo a GPSD.
Primero debemos configurar la app antes de poder enviar los datos.
Para esto abrimos la app y vamos a la seccion que dice “Connections”. Aquí tocaremos en los tres puntos que se encuentran arriba a la derecha y marcaremos la opcion “Create NMEA”. Esto asegurará que el stream NMEA sea enviado correctamente.

Luego iremos nuevamente a Connections y pondremos “Add”
Aquí tocaremos en “Activity” y seleccionaremos la primer opción “Share my GPS with a laptop or tablet that does not have GPS using NMEA”.
Luego en “Connection method” seleccionaremos “Use USB to send NMEA GPS to PC”

Luego en “Name” colocaremos “wardriving”
Por ultimo tocamos “Next” y seleccionaremos “OK” dejando el puerto 50000 por defecto.

Ahora debemos ir al menú de nuestro celular para activar el USB-Debugging. Para esto vamos a opciones de desarrollador (preguntarnos como hacerlo si no lo saben utilizar o activar). Una vez activo debemos ir a nuestra máquina Kali Linux y debemos establecer una conexión TCP/IP con ADB.
Para ello ejecutaremos el siguiente comando:

```
adb devices
```

Nuestro celular nos preguntará si deseamos establecer la conexión. Le damos que si.

Ahora debemos ejecutar el siguiente comando para establecer la conexión con el dispositivo:

```
adb forward tcp:20175 tcp:50000
```

Este comando está tomando el puerto 50000 del celular y enrutándolo al puerto 20175 de Kali Linux.
Podemos utilizar cualquier otro puerto, siempre y cuando respetemos el 50000 que configuramos dentro de la app.

Ahora vamos a ejecutar una serie de comandos para testear la conexión.

```
nc localhost 20175
```

Si todo salió bien, este comando nos deberia mostrar una salida NMEA en la consola.
Si esto es así, estamos listos para comenzar.

Ahora ejecutaremos el siguiente comando para iniciar el servidor GPSD:

```
gpsd -n -b tcp://localhost:20175
```

Y con esto estamos listos para configurar Kismet para consumir estos datos.

GPS Mediante adaptador GPS USB:

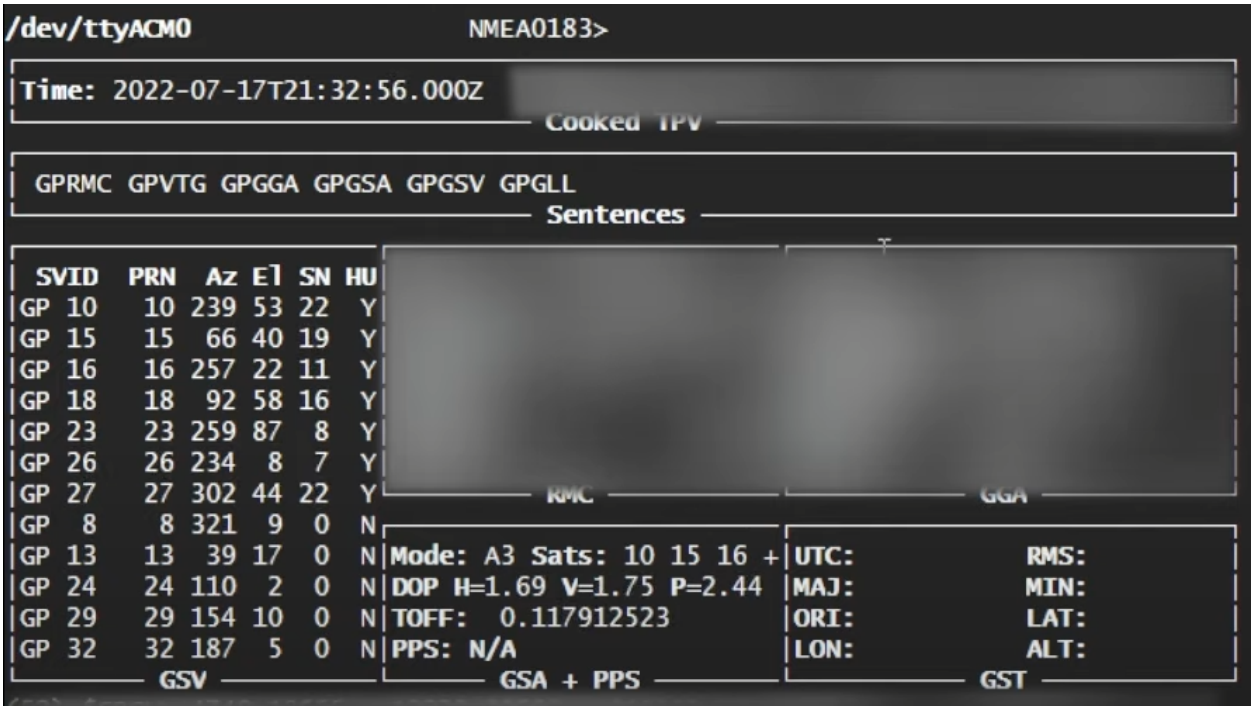
Para este método debemos contar con un dispositivo GPS compatible con GPSD.
Es muchisimo más sencillo que los otros métodos explicados más arriba, pero igual de válida.
Lo primero que debemos hacer es verificar que nuestro GPS USB esté conectado y enviando datos.
Ejecutamos:

```
sudo cat /dev/ttyACM0
```

En caso de que no sea así, debemos verificar que los kernel mods pertenecientes a su GPS estén instalados y activos.

Ahora iniciamos gpsd con este dispositivo. Para ello escribimos y ejecutamos:
`gpsd /dev/ttyACM0`

Para probar la conectividad, debemos testear con una herramienta del paquete gpsd-clients llamada gpsmon. Basta con escribir su nombre en la terminal para que nos salte un resultado similar a este:



No siempre se verá así, de hecho en la mayoría de los dispositivos se verá bastante mal! Pero lo importante son las coordenadas, y que el flujo de información esté llegando correctamente. Para testearlo basta con hacer unos pasos, aunque existen métodos más eficientes que no explicaremos acá.

5 - Configurando Kismet:

Ahora que tenemos el servidor GPSD corriendo y poseemos coordenadas, debemos configurar Kismet para poder tomar los datos NMEA que estamos forwardeando.
Para ello ejecutamos:
`vim /etc/kismet/kismet.conf`

Este paso es distinto para aquellos que utilicen el servidor GPS via WLAN que para aquellos que utilicen un adaptador GPS USB o bien el celular (estos dos ultimos son

iguales).

- Para usuarios conectados a la red con el servidor GPS:
  Deberán localizar la siguiente linea comentada “gps-gpsd:host=localhost,port=2947”. La descomentamos y colocamos la IP del servidor GPS + el puerto en el que se sirve (50000) en donde dice “localhost”.
  Quedará algo así: “gps-gpsd:host=10.3.141.202:50000,port=2947”.
  Guardamos el archivo y salimos.
- Para usuarios conectados al gps mediante adaptador USB o celular:
  Deberán localizar la siguiente línea “gps-gpsd:host=localhost,port=2947” y descomentarla.
  Guardamos el archivo y salimos.

Con esto ya tenemos el GPS configurado para Kismet.

Por último, debemos configurar el tamaño del buffer para kismet. Esto es crucial, ya que cuando se releva el medio con antenas de alta ganancia, se suele desbordar el buffer causando que kismet se cierre completamente y se pierdan minutos de wardriving que pueden ser cruciales para el conteo final de redes, así como también creará un nuevo log file que debemos unir mas tarde (no nos tiren más laburo porfa xD).
Para esto debemos abrir el archivo de configuración kismet_memory.conf y modificar su parámetro. Para lograrlo ejecutamos:

```
sudo vim /etc/kismet/kismet_memory.conf
```

Navegamos hasta donde dice “ipc_buffer_kb=kb” y lo reemplazamos por “ipc_buffer_kb = 2048” para asignarle 2mb de buffer.

A continuación, modificamos una línea más: “ulimit_mbytes=ram_in_megabytes" por “ulimit_mbytes=2048” para asignarle 2gb de RAM al proceso de Kismet.

Si llega a cerrarse Kismet con el error “buffer full”, aumentamos los valores de ambas configuraciones a 4096 (4 mb de buffer y 4gb de RAM, respectivamente).

Dejamos esto como está (sin ejecutar) y pasamos a configurar las interfaces para la captura. En una sección posterior daremos el comando de ejecución de Kismet con las interfaces y el modo wardriving.

6 - Iniciando el modo monitor de la(s) NIC(s):

Para este paso debemos contar con la suite “Aircrack-ng”.
Para instalarla ejecutaremos

```
sudo apt install aircrack-ng -y
```

Una vez instalado, debemos ejecutar lo siguiente:
`sudo ifconfig`

Esto nos mostrará las interfaces de red que tenemos, debemos anotar el nombre de las wireless para su uso con airmon.
Para colocarlas en modo monitor, ejecutamos el siguiente comando (el nombre de la interfaz es de referencia):

```
sudo airmon-ng start wlan0
```

Si el comando nos solicita ejecutar “`sudo airmon-ng checkkill`” lo hacemos.
Esto detendrá los procesos que están utilizando las interfaces de red wireless.
Por cambio, si todo sale bien, la interface debería cambiar su nombre a algo así como “wlan0mon”.
Con esto ya estamos listos para ejecutar Kismet.

7 - Iniciando el wardriving:

Con nuestro servicio GPSD corriendo y las interfaces con el modo monitor habilitado, ya podemos comenzar nuestro wardriving.
Para ello debemos ejecutar kismet en modo wardriving.
Esto no es mas que un config override que desactivará la captura de paquetes (recordatorio de que capturar tráfico wifi de una red sin autorización es **ILEGAL** en la mayoría de los paises) y afinará los filtros para solo capturar los beacons Announce que dan los detalles de un Access Point, obviando las management frames y los clientes, entre otros varios tweaks y mejoras.

Para ello ejecutaremos:

```
kismet -t nombre_del_team -c interface_monitor --override wardrive
```

Explicando un poco este comando y sus flags:

`-t` nos permite elegir un nombre para el archivo de base de datos de kismet. Podemos elegir el nombre que queramos (recomendamos que uses tu nombre, nick o el nombre de tu Team a fin de identificarte más tarde si llegases a ganar).

`-c` nos permite seleccionar la interface (monitor mode) que queremos utilizar para la captura. Podemos pasarle mas de una interfaz como argumento en caso de que las poseamos.

`--override`es el override que sobre-escribe los archivos de configuración a fin de preparar a kismet para el modo de wardriving. No debemos colocar nada que no sea wardriving.

Ahora que ya tenemos lo necesario para el wardriving.
¡Disfruten!