*Wardriving Field Manual:*

*Este es un manual que tiene como objetivo el correcto desarrollo de un wardriving (o warwalking), tanto desde el punto de vista de la preparación, el software y el hardware.*

*Se advierte desde el comienzo que se recomienda estrictamente Linux, en especial Ubuntu o Kali, para replicar lo que se ve en este manual.   
Con el tiempo se irá actualizando para agregar más contenido.*

**Indice:**

**Requisitos**

Para el desarrollo del wardrive, se requiere una lista de hardware y software bien definida, ya que la misma fue testeada antes de escribir este ensayo de carácter final.

**Hardware necesario:**

* 1 Dispositivo portátil capaz de correr Linux (nativo o virtualizado).  
  Se recomiendan al menos 2gb de RAM para antenas de ganancia muy alta.
* 1 (o más) NIC capaz de funcionar en Monitor Mode y de Packet Injection.  
  Es importante que cuente con 1 o más conectores tipo RP-SMA para adicionarle antenas con mayor ganancia.
* Cables USB preparados para datos y celular.

**Hardware opcional:**   
Este hardware no es estrictamente necesario, pero utilizarlo marca una diferencia enorme en los resultados y eficiencia del viaje.

* 1 Dispositivo GPS con interfaz USB o Serial. (Puede ser un Android, se explicará más adelante).
* 1 (o más) NIC que soporte MIMO (802.11n) y 5.8gHz (802.11a/c) y conectores RP-SMA, ademas de Monitor Mode y Packet Injection.
* 1 antena Yagi, de Placa o Parabólica (Direccional) con terminación RP-SMA o Pigtail.
* 1 antena omnidireccional de 12dBi (o más)
* 1 Celular corriendo Wiggle

**Software necesario:**Se vuelve a mencionar que este manual está pensado en Linux Debian-like.

* Sistema Operativo Linux de preferencia, virtual o nativo.   
  Los que fueron probados para este manual, al día de hoy: Ubuntu 20.04, Ubuntu 18.04, Kali Linux 2020.2a, Parrot Linux.  
  Se supone que debería funcionar en todas las distribuciones Debian sin ningún problema, pero esto está sujeto a revisión y será actualizado a medida que se hagan las pruebas. Paciencia por favor.
* (Usuarios Android): Wiggle WiFi.   
  Es una aplicación estrictamente para wardriving, con todo lo que un wadriver necesita: escanea WiFi, Bluetooth y antenas GSM. Además concatena todos estos datos con el GPS del celular.
* Kismet  
  Es LA aplicación de monitoreo Wireless 802.11x  
  sudo apt install kismet

**Software opcional:**

* (Usuarios Android): Aplicación ShareGPS.   
  Aún no hay una solución en iOS porque no hay disponible, al momento, un dispositivo físico Apple para probarlo.  
  Se agradecería muchísimo la colaboración de algún usuario Apple iPhone que encuentre una forma de utilizar el GPS del mismo vía USB a través del protocolo NMEA.
* Airodump-ng  
  Parecido a kismet, pero más básico.
* (Usuarios GPS o Android): GPSD y GPSD-Client (issues con ShareGPS)  
  sudo apt install gpsd gpsd-client
* (Usuarios GPS o Android): Netcat (nc)  
  Si bien ya viene por defecto en la mayoría de las distros de Linux, es conveniente mencionar que se utiliza para testear la conectividad del GPS de Android con la PC.
* (Usuarios Android): ADB (Android Debug Bridge)  
  Necesario para utilizar el GPS de nuestro celular con la PC via USB.

# **Preparando la salida:**

Bien, ahora que ya sabemos que es lo necesario, vamos a ver como funciona todo ello junto.  
Lo primero es la elección del sistema operativo.  
Como ya se mencionó reiteradas veces, este manual está pensado para usuarios de Linux por el momento, y fue testeado principalmente sobre Ubuntu 20.04 y Kali Linux 2020.  
Esto no quiere decir que no podamos hacerlo con Windows u OsX, pero estaríamos muy limitados en cuanto a las opciones de software para escaneo.   
  
Lo segundo es la elección del hardware de red, y esto es sumamente importante ya que no todos los NIC 802.11x son compatibles con los modos de operación que requerimos de ellos.   
Para esto necesitamos un NIC con chipset que sea compatible con modo monitor e inyección de paquetes (para passive y active scanning, respectivamente).  
A continuación se da una breve lista de chipsets compatibles:

* Atheros AR5000
* Atheros AR5001A-AR5007EG
* Intel PRO/Wireless 2100B
* Intersil ISL3877, ISL3880,ISL3890, ISL3886
* Ralink RT2770, RT2870, RT2070, RT3070, RT3071, RT3072, RT3370, RT3572, RT5370, RT5572,RT8070
* Para más información acerca de Chipsets compatibles, visitar los siguientes links:  
  <https://wifivisit.blogspot.com/2019/07/Monitor-Mode-Supported-WiFi-Chipset-Adapter-List.html>  
  <https://wifivisit.blogspot.com/2020/06/best-kali-linux-wifi-adapter-2020-india-wifi-adapter-for-kali-linux.html>

Actualmente uno de los mejores equipos para estas tareas es la placa Alfa AWUS-A360H.  
Aunque no cuenta con soporte MIMO (802.11n).  
Otro modelo es el TP-Link TL-722N en su versión v1.0. Pero tampoco tiene soporte MIMO.  
Por último, el que si tiene soporte MIMO y es económico es el NIC Edup EP-MS8515GS.

# **Instalando el software necesario.**

### Kismet

Kismet es un software de captura de señales inalámbricas que van desde WiFi a RF y BLE.   
El mismo es compatible con un número muy amplio de chipsets, sistemas operativos (aunque Linux es su preferido) y aplicaciones 3rd party para extender su funcionalidad (como GPSD).   
El mismo genera varios tipos de dump files los cuales pueden ser consultados por un sin fin de scripts o aplicaciones debido a lo simple de su formato.   
Por último, Kismet realmente no es un solo programa, si no que es el conjunto de un cliente (web o por terminal) y un servidor.   
Aunque generalmente haremos correr a ambos en el mismo anfitrión, es importante aclarar esto.   
  
Lo primero que deberíamos hacer es instalar Kismet en nuestra computadora.   
Para ello, tenemos dos caminos: o lo bajamos del repositorio oficial o lo compilamos nosotros desde la fuente.

**Para la compilación desde fuente**  
Normalmente basta con ejecutar una serie de comandos en la terminal:

1. *sudo apt install build-essential git libmicrohttpd-dev pkg-config zlib1g-dev libnl-3-dev libnl-genl-3-dev libcap-dev libpcap-dev libnm-dev libdw-dev libsqlite3-dev libprotobuf-dev libprotobuf-c-dev protobuf-compiler protobuf-c-compiler libsensors4-dev libusb-1.0-0-dev python3 python3-setuptools python3-protobuf python3-requests python3-numpy python3-serial python3-usb python3-dev librtlsdr0 libubertooth-dev libbtbb-dev*
2. *git clone https://www.kismetwireless.net/git/kismet.git*
3. *cd kismet*
4. *git pull*
5. *cd kismet*
6. *./configure*
7. *make   
   En caso de querer acelerar las cosas y contar con mas de 1 CPU Core  
   make -j$(nproc)*
8. *sudo make suidinstall*
9. *sudo usermod -aG kismet $USER*

De esta forma ya deberíamos tener una instalación fresca y completa de Kismet en nuestro S.O.   
Se incita a que todos los usuarios lo hagan de esta manera, ya que es la forma más controlada, actualizada y completa para instalar Kismet.

Para instalar desde los paquetes prearmados de kismet visitar el link y buscar. Hay muchísimas distros incluidas:

<https://www.kismetwireless.net/docs/readme/packages>  
O bien podemos instalar el paquete estándar desde bash:

* sudo apt install kismet

### Instalando GPSD:

GPSD es una aplicación para Linux, la cual nos permite leer un dispositivo GPS conectado de forma serial o por USB al host donde debería estar corriendo.   
GPSD es compatible con Kismet, el cual al iniciarse intenta comunicarse automáticamente con este.  
Esta combinación de programas nos permite la creación de un mapa de redes WiFi, concatenando las coordenadas que recibe del GPS con los datos de nuevas redes que captura Kismet.   
El resultado es un mapa del WiFi del área recorrida, con lujo de información y detalles.

### Instalando Wiggle:

Wiggle es una aplicación Android la cual nos permite hacer un pequeño wardrive sin necesidad de configurar uno o varios dispositivos, bajar controladores o software complejo de ensamblar.   
Simplemente vamos al PlayStore de nuestro dispositivo móvil Android, buscamos Wiggle WiFi y lo descargamos.  
Wiggle funciona leyendo el GPS, el Bluetooth, el GSM y el WiFi del celular, por lo cual nos va a pedir todos estos permisos.  
Además, esta aplicación cuenta con una página web la cual nos muestra las redes de todo el mundo que la gente va subiendo de forma anónima o con un usuario creado.   
La ventaja de tener un usuario en wiggle es que podemos ver los archivos que vamos subiendo, mergearlos y crear un propio mapa, aun cuando no contamos con los datos encima.

### Instalando Android Debug Bridge y ShareGPS (solo usuarios Android DUH!)

Antes de continuar con esta sección, debemos aclarar que tuvimos hasta el día de hoy algunos problemas para hacer funcionar el GPS de Android con GPSD correctamente.   
Los datos llegan correctamente a la PC, pero el GPSD server no logra recolectarlos.  
Android Debug Bridge o ADB, es una herramienta muy poderosa para configuración y operaciones avanzadas o de bajo nivel con dispositivos Android y una computadora con interfaz USB.   
En este caso, vamos a reducir esta poderosísima herramienta a una especie de enrutador entre la PC y nuestro dispositivo móvil.  
El motivo de esta instalación está apuntado solamente a usuarios que deseen hacer uso de un GPS durante la captura de redes, pero no posean un equipo dedicado a ello como un Garmin.   
Lo que se debe realizar ahora, en distribuciones Debian, es ejecutar en la consola el siguiente comando:  
 sudo apt install adb

Con esto finalizamos la instalación del mismo.  
Ahora, una vez instalado, es importante que coloquemos el dispositivo Android que vamos a utilizar en modo desarrollador.   
Para esto, el procedimiento suele ser muy parecido siempre:

1. Bajar el menú desplegable de Android e ir a opciones.
2. Ir hasta la opción del menú que diga “Acerca del dispositivo” o similares.   
   Generalmente se encuentra abajo de todo en el menú de opciones.
3. Aquí aparece un menú que muestra, entre varias otras, la opción “información del software”.  
   Presionamos aquí y nos llevará a otro menú
4. Aquí podremos visualizar “número de compilación” el cual debemos apretar reiteradas veces hasta que nos salte una notificación que dice “Ya eres desarrollador”. El cual agregará en el menú del paso 1 el submenú “opciones de desarrollador”.
5. Entramos en dicha sección y colocamos “depuración USB” como activada.
6. Conectamos el celular por el USB.

Ahora procedemos a instalar ShareGPS en nuestro celular.  
Para esto debemos hacer algo tan sencillo como con Wiggle: ir al PlayStore e instalar la aplicación con el nombre tal cual se muestra acá.  
Una vez instalada, se debe colocar el cable USB en nuestro dispositivo android, abrir la una terminal de bash y escribir

$ adb devices

Este comando nos devolverá la lista completa de dispositivos con los que adb podría comunicarse.   
A la vez, al ejecutar este comando, en nuestro dispositivo aparecerá una notificación solicitándonos permitir o denegar el acceso al dispositivo por parte de adb y una llave criptográfica del mismo.  
Cuando esto sucede