

Configuration Guide

Version: 1.0

Author: PixelatedDwarf

Purpose: Step-by-step setup for Kismet, gpsd, Python environment, and map customization on Kali Linux.

Part 1: System Preparation

Step 1: Update Kali

Before installing anything, update your system.

1. Open a Terminal.
2. Run: `sudo apt update && sudo apt upgrade -y`
3. Reboot if prompted.

Step 2: Install Python Dependencies

The toolkit scripts require Python 3 and several libraries.

1. Navigate to your toolkit folder: `cd ~/battlespace_recon`
2. Run: `pip3 install -r requirements.txt`
3. This installs Folium, Pandas, and other required packages.

Step 3: Install gpsd

GPS data is essential for geotagging your detections.

1. Run: `sudo apt install gpsd gpsd-clients -y`
 2. This installs the GPS daemon and client tools.
-

Part 2: GPS Configuration

Step 1: Connect Your GPS Device

If using a USB GPS dongle, plug it in now.

If using a smartphone, you need an app that outputs NMEA data over USB. On Android, look for "Share GPS" or "GPS2IP." On iOS, look for "GPS NMEA Stream."

Step 2: Verify the Connection

1. Run: `lsusb` to confirm the device is recognized.
2. Run: `sudo systemctl start gpsd`
3. Run: `cgps -s`
4. Wait for a satellite lock. You should see latitude and longitude updating. If you see "No Fix," move outside or near a window.

Step 3: Configure gpsd for Auto-Start

1. Run: `sudo systemctl enable gpsd`
2. This ensures gpsd starts automatically on boot.

Step 4: Set the GPS Device Port

By default, gpsd listens on `localhost:2947`. If your GPS device is on a specific serial port (e.g., `/dev/ttyUSB0`), you may need to configure it.

1. Run: `sudo nano /etc/default/gpsd`
 2. Set `DEVICES="/dev/ttyUSB0"` (or whatever port your device uses).
 3. Save and exit.
 4. Restart: `sudo systemctl restart gpsd`
-

Part 3: Kismet Configuration

Step 1: Install Kismet

Most Kali installations include Kismet. If not:

1. Run: `sudo apt install kismet -y`

Step 2: Verify Your Adapter

1. Plug in your **ALFA AWUS036AXM**.
2. Run: `iwconfig`
3. Note the interface name (usually `wlan1` or `wlx` followed by the MAC address).
4. Confirm it says "Mode: Managed" (we will switch it to Monitor later).

Step 3: Launch Kismet with GPS

1. Run: `sudo kismet`
2. In the Kismet web interface (opens in your browser at `http://localhost:2501`):
 - Click "Sources" and add your ALFA adapter.
 - Set the source type to "Linux Wi-Fi" or "RTL8812AU" depending on your driver.
 - Under "GPS," select "gpsd" and set the host to `localhost:2947`.
3. Click "Start."

Step 4: Verify GPS in Kismet

1. Look at the bottom of the Kismet interface.
2. You should see GPS coordinates updating. If it says "No GPS," revisit Part 2.

Step 5: Kismet Log Settings

By default, Kismet saves logs to `~/ .local/share/kismet/`. Each session creates a new `.kismet` SQLite database file. You will convert these to CSV using the export tools.

Part 4: Map Customization

The map generator uses a configuration file called `config.json` to define icons, colors, and popup text. This is where you make the map yours.

Step 1: Locate the Config File

Navigate to: `~/battlespace_recon/config/config.json`

Step 2: Understand the Structure

The file is divided into sections. Here is what each controls.

Icon Definitions: Each vendor or category can have a custom icon. The format uses Font Awesome icon names.

- `flock_icon`: Default is "video". Change to any Font Awesome icon name.
- `ring_icon`: Default is "bell".
- `municipal_icon`: Default is "building".
- `unknown_icon`: Default is "question".

Color Definitions: Each confidence level has a color.

- `high_confidence_color`: Default is "red". This marks nodes where SIGINT and OSINT both confirm a camera.
- `silent_node_color`: Default is "orange". This marks OSINT-only nodes (hardwired cameras).
- `private_node_color`: Default is "yellow". This marks SIGINT-only nodes (unregistered cameras).
- `baseline_color`: Default is "lightgray". This marks all other networks in the electronic terrain.

Popup Fields: These control what information appears when you click a marker.

- `show_vendor`: Default is "true". Displays the vendor name (e.g., "Ring").
- `show_signal`: Default is "true". Displays the signal strength.
- `show_source`: Default is "true". Displays the data source (SIGINT or OSINT).
- `show_timestamp`: Default is "false". Toggle to "true" if you want the detection time.

Step 3: Edit the Config

1. Open the file in any text editor: `nano ~/battlespace_recon/config/config.json`
2. Modify the values to your preference.
3. Save and exit.

Step 4: Apply the Config

When you run the map generator, point it to your config file:

1. Run: `python3 map_generator.py -s threat_view.csv -c config/config.json -o threat_map.html`
2. Open the resulting HTML file. Your custom icons and colors should appear.

Part 5: Troubleshooting

`gpsd` shows "No Fix."

- Move outside. GPS needs a clear view of the sky.
- Check the serial port. Run `dmesg | grep tty` to see which port the GPS is on.
- Restart gpsd: `sudo systemctl restart gpsd`

Kismet does not see the adapter.

- Unplug and replug the ALFA adapter.
- Run `dmesg | tail` to check for USB errors.
- Try a different USB port. The ALFA adapter needs USB 3.0 for stable power.

Kismet shows "No GPS."

- Confirm gpsd is running: `sudo systemctl status gpsd`
- Confirm Kismet is pointed to `localhost:2947`.
- Run `cgps -s` in a separate terminal to verify gpsd has a fix.

Map generator throws a Python error.

- Confirm you installed dependencies: `pip3 install -r requirements.txt`
- Confirm your CSV has the expected column headers (SSID, MAC, Lat, Lon, Vendor).
- Check the sample data files for the correct format.

Custom icons do not appear.

- Confirm the icon names are valid Font Awesome icons. Browse the Font Awesome website for a list.
 - Confirm the `config.json` file path is correct in your command.
-

Part 6: Quick Reference Commands

Start gpsd: `sudo systemctl start gpsd`

Check GPS fix: `cgps -s`

Launch Kismet: `sudo kismet`

Export Kismet to CSV: `kismetdb_to_csv -i input.kdb -o output.csv`

Run OUI Lookup: `python3 ouilookup.py -i input.csv -o output.csv`

Run Map Generator: `python3 map_generator.py -s input.csv -c config/config.json -o map.html`