Ravage-Classic

If on windows:

Install Ubuntu 24.04.1 on WSL

sudo apt update && sudo apt upgrade
sudo apt install gnome-terminal

Create a new directory to hold the necessary repos

Mkdir ~/src && cd ~/src

Next clone the ravage repo

git clone https://github.com/DependableSystemsLab/ravage.git

Then we need to clone the two autopilot repositories (stay in the src repo don't cd into ravage yet)

git clone --recurse-submodules https://github.com/ArduPilot/ardupilot.git

git clone https://github.com/PX4/PX4-Autopilot.git --recursive

Next we need to do the setup for the autopilot softwares

cd ardupilot/

./Tools/environment_install/install-prereqs-ubuntu.sh -y

cd ..

cd PX4-Autopilot/

./Tools/setup/ubuntu.sh

Next Ravage requires some configuration of environment variables to know where we put everything, the below bash script when run outputs exactly what you need to copy into the autopilot_config.yml file

```
RAVAGE_HOME=$(realpath ~/src/ravage)
ARDUPILOT_HOME=$(realpath ~/src/ardupilot)
PX4_HOME=$(realpath ~/src/PX4-Autopilot)
echo "RAVAGE_HOME=$RAVAGE_HOME"
echo "ARDUPILOT_HOME=$ARDUPILOT_HOME"
```

echo "PX4_HOME=$PX4_HOME"

Copy these paths and then cd into the ravage directory
In the ravage directory

vim config/autopilot_config.yaml

Near the top you will see something like this

```
# Main_config.yaml
#Path
RAVAGE_HOME: "/home/ravage/ravage/"
ARDUPILOT_HOME: "/home/ravage/ardupilot/"
PX4_HOME: "/home/ravage/PX4-Autopilot/"
```

Change these paths to the paths outputted by the script above

Now we need to make one more modification to get the Mavlink Communication working on WSL

While still in the ravage directory

vim open_sim.py

Make the following changes to lines 48 and 50 (these changes are now pushed to the github repo and no longer need to be made manually)

```
47  47      if wipe_eeprom == "off" and SOFTWARE_VER == "ArduPilot":
48      -       c = SOFTWARE_HOME + 'Tools/autotest/sim_vehicle.py -v ' + vehicle_target + ' --console --map'
    48  +       c = SOFTWARE_HOME + 'Tools/autotest/sim_vehicle.py -v ' + vehicle_target + ' --console --map' + ' --out=udp:127.0.0.1:14550'
49  49      elif wipe_eeprom == "on" and SOFTWARE_VER == "ArduPilot":
50      -       c = SOFTWARE_HOME + 'Tools/autotest/sim_vehicle.py -v ' + vehicle_target + ' --console --map -w'
    50  +       c = SOFTWARE_HOME + 'Tools/autotest/sim_vehicle.py -v ' + vehicle_target + ' --console --map -w' + ' --out=udp:127.0.0.1:14550'
51  51      else:
52  52          # c = 'cd ' + SOFTWARE_HOME + ' && make px4_sitl ' + vehicle_target
53  53          c = 'cd ' + SOFTWARE_HOME + ' && make px4_sitl jmavsim'
```

One more change needs to be made to your environment for ardupilot to properly launch

Run this command after opening a terminal or you can add it to the end of your .bashrc if you don't want to think about it

export $(dbus-launch)

Finally, how to actually run ravage

While in the ~/src/ravage directory

Python mission.py -s ArduPilot

This should open up a bunch of windows, mission.py will run two scripts using gnome-terminal, the first opens the simulator and the second sends the mission plan (this whole process takes 20-30 seconds so give it a bit of time to complete) After the mission has begun and the drone starts moving down its path on the graph we can launch an attack

Again from the ravage directory

python ravage.py -s ArduPilot -a GPS -i 3 -f constant_deviation

This injects a fault into the GPS system with severity three, the console should tell you when the fault starts injecting and you should see a change in the drones behavior

Ravage-Extended Instructions

Instructions are the same except change the url for the ravage repo you clone into ~/src/

Instead of doing *git clone* *https://github.com/DependableSystemsLab/ravage.git*

Do

git clone git@github.com:kyle-mc02/Ravage-Extended.git

To run ravage with the new fault types

New flag -f where you can add { precision_damage | stuck | short_circuit | drift-pos | drift-neg }

Example commands for done using one sensor, all other flags are described in the original ravage readme

```
python ravage.py -s ArduPilot -a GPS -i 5 -t 30 -f precision_damage
python ravage.py -s ArduPilot -a GPS -i 5 -t 30 -f stuck
python ravage.py -s ArduPilot -a GPS -i 5 -t 30 -f short_circuit
python ravage.py -s ArduPilot -a GPS -i 5 -t 30 -f drift-pos
python ravage.py -s ArduPilot -a GPS -i 5 -t 30 -f drift-neg
```