Noise-Orchestra

A multi-regional, multi-protocol privacy orchestra that drowns out tracking signals

---

Overview

The **Noise-Orchestra** package turns a single noise-flood script into a full-blown privacy orchestra. It ships with:

- **Geo-Clone agents** – up to 30 self-contained "virtual users", each anchored to a different country.
- **Synthetic DNS churn**, **BLE beacon broadcasting**, **mock-GPS jitter**, **MQTT IoT telemetry**, and **decoy-email** generation.
- A **lightweight scheduler** (cron / systemd) that runs the whole pipeline on a realistic daily routine (weather → news → brand sites → neutral "fluff" URLs).

All components are written in Python (with optional Node/Puppeteer for the browser part) and can run on any modern laptop, Raspberry Pi, or headless VM. The orchestra stays comfortably under **5 % of a typical home ISP data cap** while drastically lowering the signal-to-noise ratio (SNR) for any collector.

Prerequisites

Platform	Required software	Install command
Linux	Python 3.9+, pip, git, curl, `systemd` or `cron`	See the *Platform-specific* section below
macOS (Catalina +)	Homebrew, Python 3.9+, git, curl	`brew install python git curl`
Windows 10 +	Python 3.9+, Git for Windows, PowerShell, optional **WSL** for Linux-style scripts	Download from https://www.python.org/ and https://git-scm.com/

NOTE – The orchestration works with any VPN that provides a country-specific exit (WireGuard or OpenVPN). A free residential-VPN trial is sufficient for testing; for production use a paid provider with a no-logs policy (e.g., Mullvad, IVPN).

Installation

1. **Download & unzip**

```
wget https://example.com/noise-orchestra.zip   # replace with the actual URL
unzip noise-orchestra.zip -d noise-orchestra
cd noise-orchestra
```

2. **Create a Python virtual environment** (recommended)

```
python3 -m venv .venv
source .venv/bin/activate        # Linux/macOS
```

```
.\.venv\Scripts\activate          # Windows PowerShell
```

3. **Install Python dependencies**

```
pip install -r requirements.txt
```

`requirements.txt` contains:

```
requests
selenium
playwright
paho-mqtt
bleak
faker
```

If you prefer Puppeteer (Node) instead of Playwright, run:

```
npm ci                        # installs the exact versions from package-lock.json
```

4. **Set up the VPN / Tor exit for each region**

Option A – Residential VPN

- Create a folder `vpn_profiles/` and drop one OpenVPN/WireGuard config per country
- Ensure the VPN client (`openvpn` or `wg-quick`) is installed:

```
sudo apt install openvpn wireguard   # Debian/Ubuntu
```

Option B – Tor

- Edit `torrc` and add an `ExitNodes` line for each region, e.g.:

```
ExitNodes {us}
StrictNodes 1
```

- Start Tor with a custom config per region:

```
tor -f torrc_us &
```

5. **Populate region-specific config files**

The repository ships with sample files for each region. Copy the template and edit the placeholders:

```
cp config/template/* config/uk/
# edit the files:
nano config/uk/urls.txt               # list of news / weather sites for the UK
nano config/uk/ua.txt                 # user-agents for the UK clone
nano config/uk/chaff_domains.txt      # neutral political fluff URLs
```

Repeat for every region you intend to run (e.g., `config/us/`, `config/sg/`, …).

6. **Create a watchdog script (optional but recommended)**

The watchdog caps total outbound bytes to stay under 5 % of your monthly cap.

```
cp scripts/watchdog_template.sh scripts/watchdog.sh
```

```
chmod +x scripts/watchdog.sh
nano scripts/watchdog.sh          # set YOUR_MONTHLY_CAP_GB
```

Running a Geo-Clone

The main entry point is `geo_clone.sh`. It expects two arguments: **region code** (e.g., `uk`) and **clone ID** (e.g., `03`).

```
./geo_clone.sh uk 03
```

What the script does (in order):

1. **Starts the VPN/Tor** for the chosen region.
2. **Exports the proper timezone & locale** (`TZ=Europe/London`, `LANG=en_GB.UTF-8`).
3. **Launches the headless browser** with its own profile directory (`profiles/uk_03`).
4. **Runs the daily routine** – weather → news → brand site → chaff URLs → optional travel search.
5. **Fires synthetic DNS queries** (via `dns_junk.py`).
6. **Publishes one IoT telemetry packet** to the region-specific MQTT topic.
7. **Broadcasts a BLE beacon** for 5 minutes (`ble_beacon.py`).
8. **Logs a one-line summary** to `dust_activity.log`.
9. **Shuts down the VPN** and exits.

Scheduling the clones

You can use **cron** or **systemd timers**. Below is a sample `crontab` entry that runs three clones during their local "working hours" (08:00-20:00).

```
# m h  dom mon dow   command
0 8-20 * * * /home/user/noise-orchestra/geo_clone.sh uk 01 >> /home/user/noise-orchestra/dust_activity.log 2>&1
15 9-21 * * * /home/user/noise-orchestra/geo_clone.sh us 02 >> /home/user/noise-orchestra/dust_activity.log 2>&1
30 7-19 * * * /home/user/noise-orchestra/geo_clone.sh sg 03 >> /home/user/noise-orchestra/dust_activity.log 2>&1
```

For **systemd**, copy `systemd/geo_clone@.service` and `systemd/geo_clone@.timer` into `/etc/systemd/system/`, enable the timer for each clone, and start the service:

```
sudo cp systemd/geo_clone@.service /etc/systemd/system/
sudo cp systemd/geo_clone@.timer   /etc/systemd/system/
sudo systemctl daemon-reload

# Enable timer for UK clone 01 (runs at 08:00 local UK time)
sudo systemctl enable geo_clone@uk_01.timer
sudo systemctl start  geo_clone@uk_01.timer
```

Monitoring & Reporting

All clones write a single line to `dust_activity.log`:

```
[2026-01-13T08:12:34Z] [UK_03] BYTES=3.2MB DNS=12 MQTT=1 BLE=1
```

A small helper script aggregates the log and prints a human-readable summary:

```
./scripts/report_snr.sh
```

Typical output:

```
=== Weekly Noise Report ===
Total outbound: 4.7 GB (4.9 % of 100 GB cap)
Average SNR: 0.14   (lower is better)
Active regions: UK, US, SG, KE, NZ
```

Contributing

1. Visit Pixelateddwarf.com and comment on the articles.

License

The Noise-Orchestra code is released under the **MIT License**. See `LICENSE` for full terms.

Quick-Start Cheat-Sheet

```
#  Extract
wget https://example.com/noise-orchestra.zip
unzip noise-orchestra.zip && cd noise-orchestra

#  Virtualenv + deps
python3 -m venv .venv && source .venv/bin/activate
pip install -r requirements.txt

#  Configure a region (example: UK)
cp config/template/* config/uk/
nano config/uk/urls.txt   # edit to your taste

#  Run a single clone (test)
./geo_clone.sh uk 01

#  Schedule (cron example)
crontab -e   # paste the three lines from the Scheduling section
```

You’re now ready to generate a **global, multi-protocol noise cloud** that makes profiling significantly harder while staying well within your data budget. Enjoy the privacy orchestra!