

## Social Engineering Location Tracker

A production-ready tool for capturing geolocation data through social engineering techniques. It generates HTTPS links disguised as photo, video, or song content to trigger browser geolocation prompts. The tool features a modern, dark-themed Tkinter GUI, Nginx with Let's Encrypt for secure hosting, and Gunicorn for scalable Flask server deployment.

Ethical Use Only: This tool is for educational purposes or authorized testing only. Use requires explicit user consent or legal authorization. Unauthorized tracking is illegal and unethical.

### Features

Modern GUI with dark theme for intuitive operation.

Generates secure HTTPS links (e.g., [https://your\\_domain.com/photos/abc123](https://your_domain.com/photos/abc123)).

Supports photo, video, and song bait content to capture geolocation.

Displays captured coordinates in a sortable table with Google Earth links.

Saves data to JSON files with a user-friendly file dialog.

Production-ready with Nginx, Let's Encrypt, and Gunicorn.

Cross-browser compatibility (Chrome, Firefox, Safari, Brave).

Comprehensive logging and unit tests for reliability.

### Prerequisites

Before setting up the project, ensure you have:

Server: Ubuntu 20.04+ with a public IP address.

Domain: A registered domain (e.g., [your\\_domain.com](https://your_domain.com)) with DNS A record pointing to your server's IP.

Python: Version 3.8 or higher.

Media Files: Sample files (bait.jpg, bait.mp4, bait.mp3) for bait content.

Access: Sudo privileges for installing packages and configuring Nginx.

Git: For cloning the repository (optional).

### File Structure

The project is organized as follows:

```
location_tracker/
├── location_tracker.py # Main script with Tkinter GUI
├── config.py # Configuration (domain, host, port)
├── web/
│   ├── app.py # Flask server for handling links and geolocation
│   ├── templates/
│   │   ├── photo.html # Landing page for photo content
│   │   ├── video.html # Landing page for video content
│   │   └── song.html # Landing page for song content
│   ├── static/
│   │   ├── images/
│   │   │   └── bait.jpg # Sample photo file
│   │   ├── videos/
│   │   │   └── bait.mp4 # Sample video file
│   │   └── audio/
│   │       └── bait.mp3 # Sample audio file
```

```

| | |— js/
| | |  └─ capture.js # JavaScript for geolocation capture
|— utils/
| |— logger.py # Logging setup
| |— output.py # Output formatting and JSON saving
| |— link_masker.py # Masked link generation
|— tests/
| |— init.py # Test package
| |— test_gui.py # GUI tests
| |— test_web.py # Flask server tests
| |— test_link_masker.py # Link masking tests
| |— test_output.py # Output formatting tests
| |— test_geolocation.py # Geolocation capture tests
|— nginx/
|  └─ location_tracker.conf # Nginx configuration for HTTPS
|— gunicorn_config.py # Gunicorn configuration for Flask
|— requirements.txt # Python dependencies
|— location_tracker.log # Application logs
|— README.md # This file

```

## Setup Instructions

Follow these steps to set up the project on an Ubuntu server. Replace `your_domain.com` with your actual domain and `/path/to/location_tracker` with the absolute path to your project directory.

### 1. Clone the Repository

Clone the project to your server (or copy files manually):

```
git clone https://github.com/SunnyThakur25/location_tracker.git
cd location_tracker
```

### 2. Install System Dependencies

Update the system and install required packages:

```
sudo apt update && sudo apt upgrade -y
sudo apt install -y nginx python3-pip python3-certbot-nginx git
```

### 3. Install Python Dependencies

Install the Python packages listed in `requirements.txt`:

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

### 4. Configure Project Settings

Edit `config.py` to set your domain:

```
nano config.py
```

Update the `DEFAULT_DOMAIN` variable:

```
DEFAULT_DOMAIN = "your_domain.com" # Replace with your domain
```

### 5. Place Media Files

Ensure sample media files are in the correct directories:

Copy bait.jpg to web/static/images/.

Copy bait.mp4 to web/static/videos/.

Copy bait.mp3 to web/static/audio/. If you don't have these files, create placeholders or update the templates (photo.html, video.html, song.html) to use alternative URLs.

## 6. Configure Nginx

Copy the Nginx configuration file to the system:

```
sudo cp nginx/location_tracker.conf /etc/nginx/sites-available/location_tracker
```

Edit the file to replace placeholders:

```
sudo nano /etc/nginx/sites-available/location_tracker
```

Replace:

your\_domain.com with your actual domain (e.g., tracker.example.com).

/path/to/location\_tracker with the absolute path (e.g., /home/user/location\_tracker).

Enable the configuration and test Nginx:

```
sudo ln -s /etc/nginx/sites-available/location_tracker /etc/nginx/sites-enabled/
```

```
sudo nginx -t
```

```
sudo systemctl restart nginx
```

## 7. Obtain Let's Encrypt SSL Certificate

Secure your domain with a free SSL certificate:

```
sudo certbot --nginx -d your_domain.com
```

Follow the prompts to configure HTTPS. Certbot will update location\_tracker.conf with the correct certificate paths.

## 8. Open Firewall Ports

Allow HTTP and HTTPS traffic:

```
sudo ufw allow 80
```

```
sudo ufw allow 443
```

```
sudo ufw status
```

## 9. Set Up Gunicorn Service (Production)

Create a systemd service for running the Flask app with Gunicorn:

```
sudo nano /etc/systemd/system/location_tracker.service
```

Add the following content:

[Unit]

Description=Location Tracker Gunicorn Service

After=network.target

[Service]

User=your\_user # Replace with your username

WorkingDirectory=/path/to/location\_tracker

ExecStart=/usr/local/bin/gunicorn --config gunicorn\_config.py -w 4 web.app:app

Restart=always

[Install]

WantedBy=multi-user.target

Replace:

your\_user with your server username.

/path/to/location\_tracker with your project path.

Enable and start the service:

```
sudo systemctl enable location_tracker
```

```
sudo systemctl start location_tracker
```

```
sudo systemctl status location_tracker
```

### Running the Application

You can run the application in two modes: development (with GUI) or production (headless server).

#### Development Mode (With GUI)

Ensure you have a graphical environment (e.g., X11 or VNC) or run locally.

Start the GUI: `python location_tracker.py`

In the GUI:

Enter your domain (e.g., your\_domain.com).

Select content type (photo, video, or song).

Click "Generate Link" to create a masked link (e.g., [https://your\\_domain.com/photos/abc123](https://your_domain.com/photos/abc123)).

Copy the link and paste it into a browser to test.

Allow geolocation in the browser to capture coordinates, which appear in the GUI table.

Click "Open Google Earth" to view locations or "Save JSON" to export data.

#### Production Mode (Headless Server)

Ensure the Gunicorn service is running (see step 9 above).

Access the GUI from a local machine by copying the project directory and running `python location_tracker.py`.

Generate links in the GUI and distribute them for authorized testing.

Coordinates are captured and displayed in the GUI when users access the links.

### Testing

Run unit tests to verify functionality:

```
python -m unittest discover tests -v
```

The tests cover:

GUI functionality (link generation, table updates).

Flask server (link redirection, geolocation capture).

Link masking, output formatting, and geolocation logic.

### Troubleshooting

#### Links Don't Open:

Verify DNS: Ensure your\_domain.com resolves to your server IP (ping your\_domain.com).

Check Nginx: `sudo systemctl status nginx` and `/var/log/nginx/error.log`.

Geolocation Fails:

Ensure HTTPS is used (geolocation requires secure contexts).

Test in Chrome or Firefox; some browsers (e.g., Brave) may block geolocation.

Gunicorn Errors:

Check logs: `sudo systemctl status location_tracker` or `location_tracker.log`.

Certbot Issues:

Review logs: `/var/log/letsencrypt/letsencrypt.log`.

Ensure port 80 is open and DNS is correct.

Ethical Guidelines

Consent Required: Use only with explicit user consent or legal authorization.

Data Privacy: Delete captured data after use.

Legal Compliance: Unauthorized tracking is illegal in most jurisdictions.

Disclaimer: The authors are not responsible for misuse of this tool.

License

MIT License. See LICENSE file for details (not included in this structure but can be added).

Contact

For issues or questions, open an issue on the repository or contact the maintainer.

Last Updated: June 19, 2025