

Configuration Manual

ENHANCING API SECURITY WITH MIDDLEWARE-BASED SECURE API GATEWAYS

MSc Research Project
MSc in Cyber Security

Abhijith Thanippilly Soman
Student ID: 23271400

School of Computing
National College of Ireland

Supervisor: Michael Pantridge

National College of Ireland
MSc Project Submission Sheet



School of Computing

Abhijith Thanippilly Soman

Student Name:
Student ID: 23271400
Programme: MSc in Cyber Security **Year:** 2024-2025
Module: Practicum Part 2
Supervisor: Michael Pantridge
Submission Due Date: 14/09/2025
Project Title: ENHANCING API SECURITY WITH MIDDLEWARE-BASED SECURE API GATEWAYS
295
Word Count: **Page Count** 2.....

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Abhijith Thanippilly Soman
Date: 14/09/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Abhijith Thanippilly Soman
Student ID: 23271400

1 System Configurations

- System: Local machine or cloud environment
- IDE: Jupyter Notebook (for model training), VS Code / Anaconda (for Flask app)
- Python Version: Python 3.8+
- Libraries used: pandas, numpy, scikit-learn, flask, joblib, requests, bootstrap (for UI)
- File types: .csv dataset, .pkl model and scaler files, HTML templates

2 Code Explanation

2.1 Model Training (anomalyPredictorModels.ipynb):

1. Load API Access Behaviour Anomaly Dataset from Kaggle
2. Data cleaning and preprocessing (missing values, irrelevant column removal)
3. Feature scaling using StandardScaler
4. Model training:
 - One-Class SVM (trained on benign data only)
 - Tuned One-Class SVM (GridSearchCV for nu & gamma)
 - Hybrid Random Forest (benign + labelled anomalies)
5. Save models (ocsvm_model.pkl, hybrid_model.pkl) and scaler.pkl

2.2 Flask App (App.py):

1. Load models and scaler
2. /predict route — HTML form, model selection, single prediction
3. /predict_json route — JSON input, Bearer token check, bulk/API prediction
4. Return prediction as “normal” or “anomaly”

2.3 Bulk Testing:

1. json_requests.py — contains 10 sample JSON inputs
2. run_json_requests.bat — starts Flask app, sends all JSON requests automatically

3 Folder Structure

Dataset used: <https://www.kaggle.com/datasets/tangodelta/api-access-behaviour-anomaly-dataset>

```
|— anomalyPredictorModels.ipynb
|— ocsvm_model.pkl
|— hybrid_model.pkl
|— scaler.pkl
```

```
├── App.py
├── json_requests.py
├── run_json_requests.bat
├── supervised_dataset.csv
├── /templates
│   ├── index.html
│   └── result.html
```

4 Executing the Code

Install dependencies for Notebook training: `pip install notebook pandas numpy scikit-learn joblib`

Install dependencies for Flask app: `pip install flask pandas numpy scikit-learn joblib requests`

Step 1: Ensure Python 3.8+ and dependencies are installed (see installation commands above)

Step 2: Verify `ocsvm_model.pkl`, `hybrid_model.pkl`, and `scaler.pkl` exist (or run the notebook to create them)

Step 3: Start Flask:

`python App.py`

Step 4: For UI — visit `http://127.0.0.1:5000` in browser, select model, input values, submit

Step 5: For bulk — run `run_json_requests.bat` or `python json_requests.py`