

Challenge 01

RAG on the Ground

Exercise	Operation Electro Magnetic Panic
First run	17 October 2025, Madrid, Spain
Published by	303 Overwatch A.S.B.L, R.C.S. Luxembourg F13274
Licence	CC BY-SA 4.0. Free to run, adapt and share.

This brief is contributed as open, non-commercial work in support of exercise and workforce development obligations under the NIS2 Directive. It is guidance, not legal advice.

Contents

Scoring Model	2
1. Objective	2
2. Required Capabilities	3
3. What Will Be Provided to Teams	3
4. Submission Deliverable	4
5. Mentor Evaluation Process	4
6. Minimal Compliance Checklist (Mentor View)	5
7. Additional Guidance for Teams	5
8. Summary of Submission Flow	5

Challenge 1, RAG On the Ground

Scoring Model

Condition	Points
First team to submit	+10 bonus
Submission within Hour 1	+10 base
Submission within Hour 2-3	+5 base
Submission after Hour 3	0 base
Each failed mentor check (OpenAI / N2YO / RAG privacy)	-5 each
Maximum penalty	-15 points

Important: Teams can end up with **negative points** if they submit early but fail the functional checks.

1. Objective

Stand up a secure, functioning retrieval-augmented generation (RAG) application that:

- Uses OpenAI for chat and embeddings (**get your API key from Open AI make sure you understand the limitations**)
- Integrates N2YO for real-time satellite position queries (**get your API key from N2YO and make sure you understand your limitations**)
- Protects the confidentiality of sensitive documents by ensuring no document metadata is ever exposed to the end user via the RAG application

Note: A script will be provided to **expedite the baseline deployment** of the RAG application.

This script is **not a complete solution**. Teams must **review, harden, and secure** their deployments beyond the baseline to meet all challenge requirements.

2. Required Capabilities

A. Authentication and Access

- Every endpoint (upload, chat, N2YO) must be behind Basic Auth or an equivalent access control mechanism.
- All API keys must be stored **server-side only**.

B. RAG Functionality

- Accept PDF, DOCX, CSV, or plain text.
- Index only the **textual content** of uploaded files.
- Generate answers strictly from that content. If uncertain, the assistant must respond with a refusal or an uncertainty statement.
- Implement guard rails prevent exposure of **filenames**, document authors, file types, or any other metadata. (the hackers will be looking for steal this from you later)

C. N2YO Integration

- The parametric /n2yo/track and NLQ /n2yo/nlq endpoints must function correctly.
- NLQ must interpret natural language inputs and clamp excessive durations.

D. Security Requirements

- Qdrant must **not** be exposed directly to the internet.
- Basic Auth credentials must protect all user-facing routes.
- No secrets or documents should appear in browser dev tools or logs.

3. What Will Be Provided to Teams

- A Google Docs folder
 - Deployment baseline script and reference instructions

4. Submission Deliverable

When the app is fully deployed and the team believes it meets the requirements:

Create a password-protected Word document named:

team-<TEAM_NUMBER>-credentials.docx

1. Inside the document, include only:
 - Deployment URL (IP or hostname)
 - Basic Auth username and password for mentor login
 - Any optional deployment notes
2. Set a **password** for the file.
3. Submit the file to the **Google Docs challenge folder**.
 - Do **not** upload any documents to the RAG app before submission.
 - Submission marks the **official completion time**.
4. Send the **password separately** through the agreed secure channel

5. Mentor Evaluation Process

Once credentials are received, mentors will:

- Log in using the Basic Auth credentials provided.
- Upload the sensitive test documents.
- Validate:
 - OpenAI integration
 - N2YO parametric and NLQ endpoints
 - RAG privacy behavior (no metadata leakage)

If any of these checks fail, a 5-point deduction will be applied per category, even if the team was the first to submit.

6. Minimal Compliance Checklist (Mentor View)

- Authentication works (401 without creds, 200 with creds).
- OpenAI integration is functional and grounded.
- N2YO parametric endpoint operational.
- N2YO NLQ endpoint parses correctly.
- RAG app uses uploaded text only and reveals no metadata.
- Logs and client view free of sensitive information.

7. Additional Guidance for Teams

- Remove or anonymize **all metadata** during indexing.
- Adjust the system prompt to explicitly instruct the model not to reveal metadata.
- Replace filenames in app/main.py with opaque IDs or remove them entirely.
- In the UI, display only **source counts** or internal IDs.
- Remove Qdrant port exposure in docker-compose.yml.
- Keep environment variables private.
- Harden the baseline script: the provided script is **only a starting point**, not a complete defense.

8. Summary of Submission Flow

1. Deploy and verify RAG + N2YO + OpenAI integration.
2. Prevent any file metadata exposure in your RAG app.
3. Prepare a password-protected Word document containing mentor credentials.
4. Upload the file to the Google Docs challenge folder.
5. Submit the password via the secure channel.
6. Wait for mentor inspection results and scoring.