Challenge 02

Secure the Signal

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.

303 Overwatch A.S.B.L, 10 rue Willy Goergen, L-1636 Luxembourg
www.303overwatch.org • workforce.development@303overwatch.org

Contents

Challenge 2, Secure the Signal

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 / Vector DB encryption)	-5 each
PQC extra credit (hybrid TLS, PQC key exchange, PQC encrypted at-rest)	**+50 points**
Maximum penalty	-15 points

> **Important:** Teams can earn significant bonus points for PQC but can also lose points if baseline encryption checks fail.

1. Objective

Enhance the RAG application from Challenge 1 by implementing **encryption in transit and at rest** across all critical communication paths:

- App ↔ Vector Database (e.g., Qdrant),
- App ↔ OpenAI,
- App ↔ N2YO.

Teams that additionally integrate **post-quantum cryptography (PQC)** elements (for example, PQC-capable key exchange for TLS or PQC-encrypted at-rest storage) will receive **+50 points extra credit**.

This challenge shifts the emphasis from functional deployment to secure transmission and storage of sensitive data.

2. Required Capabilities

A. Encryption In Transit

- All traffic between application components must be secured with TLS:
  - Qdrant must run with TLS (or be accessed via a reverse proxy that enforces TLS).
  - All communication with OpenAI and N2YO must be verified as HTTPS traffic. >
  - TLS 1.3 is strongly recommended; TLS 1.2 is the minimum accepted version.
  - Self-signed, Let's Encrypt, or internal CA certificates are acceptable, provided they are configured correctly.

B. Encryption At Rest

- Data stored in the vector database must be encrypted at rest.
- Encryption keys and TLS private keys must not be hard-coded, committed to Git, or stored in client-side code.
- Acceptable approaches include encrypted volumes, encrypted filesystems, or integrated encryption mechanisms.

C. PQC Extra Credit (Optional)

- PQC-capable TLS handshake (e.g., hybrid TLS 1.3 with Kyber or equivalent).
- PQC-protected local storage or key exchange.
- Teams must provide evidence through configuration outputs, screenshots, or a short proof-of-concept description.

D. Security Requirements

- No plaintext traffic between components.
- No sensitive secrets or private keys exposed in logs, environment variables, or front-end code.
- TLS configurations must be stable, verifiable, and enforced.

3. What Will Be Provided to Teams

- A **Google Docs folder**

4. Submission Deliverable

When the team has completed their encryption implementation:

Create a password-protected Word document named:

team-<TEAM_NUMBER>-signal-security.docx

1. Inside the document, include:
    - Deployment URL (IP or hostname)
    - Basic Auth username and password for mentor login
    - Summary of TLS configuration (e.g., Caddy/Nginx, Qdrant cert setup, cipher suites)
    - Storage encryption configuration details (method/tool used)
    - If PQC implemented:
        - Short explanation of the approach
        - Sample code, configuration, or library reference
        - Screenshot or text output showing cipher suite or algorithm in use
2. Set a **password** for the file.
3. Upload it to the **Google Docs challenge folder**.
4. Send the password securely to the mentor.

5. Mentor Evaluation Process

Once credentials are received, mentors will:

- Log in using the provided Basic Auth credentials.
- Verify encryption in transit:
    - TLS on Qdrant connections.
    - HTTPS verification for OpenAI and N2YO calls.

- Verify encryption at rest:
  - Check storage encryption evidence.
- Attempt to detect plaintext traffic or logs.
- If PQC extra credit is claimed, review the configuration, cipher evidence, or proof-of-concept.

A **5-point deduction** applies for each failed category (OpenAI / N2YO / Vector DB encryption) regardless of submission timing.

6. Minimal Compliance Checklist (Mentor View)

- Authentication still enforced (from Challenge 1).
- TLS enabled for Qdrant communication.
- HTTPS verified for OpenAI and N2YO.
- Storage encryption is active and documented.
- No plaintext secrets or traffic visible in logs, captures, or configs.
- PQC extra credit validated (if applicable).

7. Additional Guidance for Teams

- Use valid certificates (self-signed or trusted) and configure services to **reject non-TLS traffic**.
- Enforce modern TLS versions (prefer TLS 1.3).
- Use encrypted volumes or filesystem-level encryption for at-rest protection.
- For PQC:
  - Consider libraries implementing hybrid TLS 1.3 with Kyber.
  - Document the handshake or storage configuration clearly.
  - Keep PQC implementation scoped and demonstrable-proof of concept is acceptable.

8. Summary of Submission Flow

1. Implement TLS on all internal and external connections.
2. Encrypt storage at rest.
3. Optionally add PQC elements for extra credit.
4. Prepare password-protected Word document with all required evidence.
5. Upload to the Google Docs challenge folder.
6. Send password securely to the mentor.
7. Mentor will inspect and score.

9. Extra Credit Guidelines: PQC

Teams can earn **+500 points** if PQC features are implemented and verifiable. Acceptable examples:

- TLS 1.3 handshake using Kyber or another PQC KEM.
- PQC-enabled encrypted volume or key exchange.
- Integration of external PQC libraries with demonstrable output.

Evidence must include **one or more of**:

- Cipher suite listing with PQC element,
- Configuration snippets or library references,
- Screenshots or output logs proving PQC integration.

Partial PQC implementations without verifiable evidence will not receive points.