TRUESEC

Code Challenge

Classification: TS Public

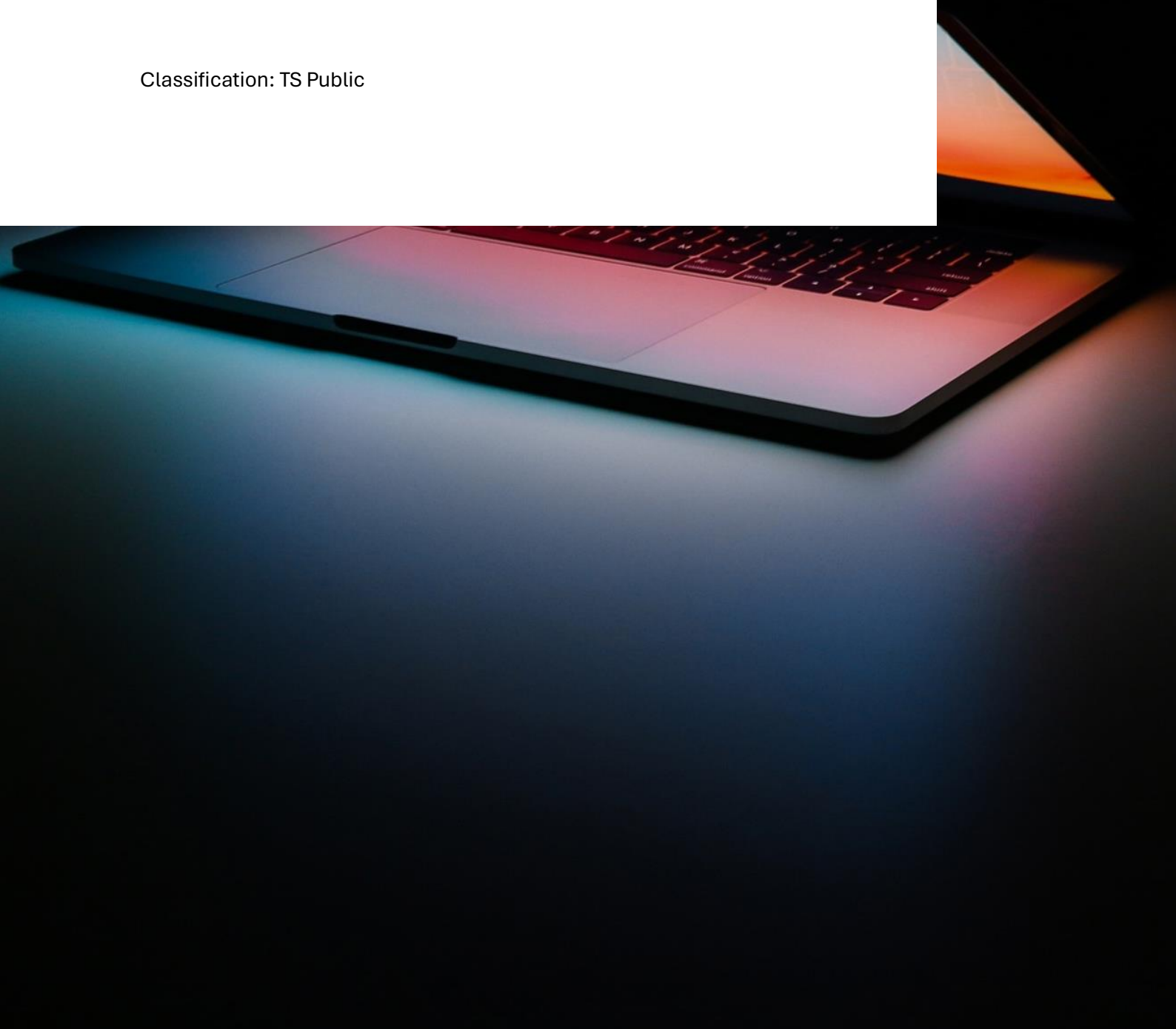

Table of Contents

Challenge

Design and implement an ASP.NET Core API that provides a RESTful interface for managing and querying vulnerabilities using the STIX II vulnerability model. The API should conform to the OpenAPI specification to ensure interoperability and ease of use.

To implement the API, you should use ASP.NET Core and the Swashbuckle.AspNetCore package to generate the OpenAPI specification. You should also use a database system such as a relational database or a document database to store the vulnerability data.

To test the API, you can use a tool such as Postman or Swagger UI to send requests and verify the responses. You should also include unit tests to ensure the correctness and robustness of the API.

API endpoints

The API should include the following endpoints:

- GET /vulnerabilities: This endpoint should retrieve a list of all vulnerabilities in the system.
- GET /vulnerabilities/{id}: This endpoint should retrieve a specific vulnerability by its ID.
- POST /vulnerabilities: This endpoint should create a new vulnerability using the STIX II vulnerability model.
- PUT /vulnerabilities/{id}: This endpoint should update an existing vulnerability by its ID using the STIX II vulnerability model.
- DELETE /vulnerabilities/{id}: This endpoint should delete an existing vulnerability by its ID.

API features

The API should also include the following features.

Input validation

The API should validate all input data to ensure that it conforms to the *STIX II* vulnerability model.

Authentication and authorization

The API should require authentication and authorization to access and modify vulnerabilities. Users should be authenticated using JWT tokens and roles should be defined to restrict access to certain endpoints.

Error handling

The API should handle errors gracefully and provide meaningful error messages to users.

Additional features

Bonus if it also has the following features.

Pagination

The GET vulnerabilities endpoint should support pagination to retrieve a subset of vulnerabilities at a time.

Filtering and sorting

The GET vulnerabilities endpoint should support filtering and sorting based on specific fields such as severity, status, and date.

Submit your solution

Create a public GitHub repository and share it with us when you are happy with your solution. Send us an e-mail and we will set up a follow-up interview with you where you present your design and solution and we will have a conversation around it.

References

- STIX model - https://oasis-open.github.io/cti-documentation/stix/intro.html
  - Vulnerability model - https://docs.oasis-open.org/cti/stix/v2.1/os/stix-v2.1-os.html#_q5ytzmajn6re
  - JSON Schema for Vulnerability can be found here: https://github.com/oasis-open/cti-stix2-json-schemas/blob/master/schemas/sdos/vulnerability.json