Hi There,

Thank you for applying for a development position in f5!

In order to continue the process, please complete and submit the attached assignment.

This assignment is designed to test programming skills, interacting and understanding existing code base, as well as the ability to learn new Concepts/Technologies and apply these in practice.

The attachments contains an implementation of a simple bank API in Go, which supports user authentication using tokens. The API includes the ability to manage users, accounts, and balances. The users can have different roles (regular users vs admin users).

Key Features:

- **User Authentication**: Token-based using JWT.
- **Role-based Access**: Admin and Regular Users.
- **Endpoints**:
    - **User**: Register, Login, Get Users (Admin only).
    - **Account**: Create Account, Get Account Details.
    - **Balance**: Get Balance, Deposit, Withdraw (Admin can view any user's balance).

Please complete the following tasks:

1. Identify and fix problems with the API implementation. (Focus on security asspects!)

   Explain (in README or with code comments) what are the problems and how did you fix.

2. Implement the main function and server parts (the server part is the code that listens to requests and serves the API, see http package)

3. Add access logging

   Log the request and response for each API call into a file

   The format should be a JSON of the following form (should output a single line in your software):

```
{
    "req":
    {
        "url": "<REQUEST_URL>",
        "qs_params": "<REQUEST_QUERY_STRING_PARAMS>",
        "headers": "<REQUEST_HEADERS>",
        "req_body_len": "<BODY_LENGTH_IN_BYTES>"
    },
    "rsp": {
        "status_class": "<STATUS_CLASS>",
        "rsp_body_len" :  <BODY_LENGTH_IN_BYTES>
    }
}
```

- You should replace placeholders with the values taken from the actual request/response.
- STATUS_CLASS is either “2xx”, “3xx”, “4xx” or “5xx” depending on the response status code. For example, if the response status code was 301, STATUS_CLASS should be “3xx”.

4. Write a program that gets an access-log file-name as input and detects potential BOLA attacks. You can assume that file format is valid and that the entire file fits in memory.
5. Upload the API server code to a public git repository.

   Upload the BOLA detection tool to another public git repository.

   Reply to email with both links.

Useful links:

https://docs.docker.com/get-started/

https://owasp.org/API-Security/editions/2023/en/0x11-t10/

https://go.dev/doc/tutorial/getting-started

https://owasp.org/API-Security/editions/2023/en/0xa1-broken-object-level-authorization/

Good Luck! Enjoy and take the time to learn ☺

- API Security Team