Source Code leakage

Monitoring Source Code Leakage

Monitoring platforms such as GitHub, GitLab, Postman, and SwaggerHub for sensitive credentials is essential to prevent unauthorized access and potential data breaches. Here's a structured approach to monitoring these platforms for source code leaks:

1. **GitHub & GitLab Monitoring**

- **Set Up Regular Scans**:
    - Periodically scan all public and private repositories for any potential hardcoded credentials such as API keys, tokens, and passwords.
    - Ensure both historical and new commits are included in the scan to capture any past leaks.
- **Monitor Repository Activity**:
    - Track repository changes and ensure that any sensitive files, such as `.env`, `config`, or `credentials`, are not being committed unintentionally.
- **Integrate Security Checks**:
    - Implement pre-commit hooks to flag sensitive information before code is pushed to repositories.
    - Enable repository secret scanning features to identify potential exposures in real time.

2. **Postman Monitoring**

- **Monitor API Collections**:
    - Ensure that API keys, tokens, and secrets are not stored in Postman environments or within API request bodies.
    - Set up automated scans for Postman collections to identify any inadvertent exposure of credentials in requests.
- **Secure Environments**:
    - Enforce the use of encrypted environments in Postman to prevent the accidental leakage of sensitive data such as API secrets.
    - Limit access to critical environments and set role-based permissions for teams.

3. **SwaggerHub Monitoring**

- **Monitor API Documentation**:
    - Ensure that sensitive information like access tokens, credentials, or keys is not exposed in API documentation published on SwaggerHub.

- Regularly scan published Swagger or OpenAPI specifications for any hardcoded secrets or sensitive data.