DSEC: A Data Analyzer tool for ensuring secure software development life-cycle

Sponsor

Vikrant Kaulgud, Accenture Labs, Bangalore, India.
Kapil Singi, Accenture Labs, Bangalore, India.

Project Abstract

Data is a critical and differentiating factor for organizations. Government and lawmakers introduced policies and regulations to govern the usage, storage, and processing of this data to safeguard user rights and allow its ethical consumption. Various policies such as GDPR, White House executive order on cybersecurity, zero trust architecture etc. highlight security practices and guidelines to mitigate unauthorized access and storage of PII data. The aim of this project is to help developers to find security hotspots and inconsistencies in different SDLC artifacts to suit the policies governing data.

Project Description

As the scope of the analysis can be large in the entire SDLC cycle, we are focusing on understanding inconsistency between various SDLC phases in the ecosystem such as

1. Requirement Phase: Minimizing data capture (redundant/additional fields/permissions) when designing and development of the service to limit the overhead of maintaining irrelevant (limited to what is necessary in relation to purpose of data capture) data.
2. Build/Deploy Phase: Ensuring sensitive data out of logs to mitigate security breaches. Identification of sensitive data being pushed in logs leveraging static and dynamic analysis of the code. Potential areas of infringement being
    a. *API Calls*: Sensitive data as request params in URL,
    b. *Input, Output and Log Statements*: Excessive use of loggers and output statements for debugging in code as well as the libraries in use.
    c. *Code and Artifacts (Config)*: Reference to personal data being used for creation of unique identifiers in code and comments, improper hashing/anonymizing the sensitive information as well as config files (Json, db connections) with static values on personal information.
    d. *Deployment Logs*: Identification in containers and deployment tool logs with DEBUG level alert emitting personal information.
3. Test Phase: Screening client and sensitive data for test cases.
        a. *Live Data Inclusion*: Identification and separation of live client data being used for testing of AI models with proper technical measures to anonymize data to ensure level of security appropriate to the risk.
        b. *Sensitive Data Artifacts for Functional Testing:* Identification and remediation to limit use of test artifacts actual or synthetic data with proximity to real and sensitive information for functional testing.

Project Scope

The project needs to incorporate the features described above but scope is not limited to them. Extension and enhancement during the project flow are possible.

Project Requirements

Teams are free to choose the development process; however, this choice should be motivated and discussed in the project report. Project artifacts should be hosted on a public repository (such as GitHub), to track the project activity (including bug reports and bug fixes). The project should be suitable for adoption and extension by other developers by downloading or forking from the repository.

Environment Constraints

The result must be a Web Service that allows clients to remotely call functions to run the analysis on the SDLC pipeline to obtain the results and meta-data related to a software project. Teams can freely choose the development environment, language, and tools to be used to build the capability.

Project Restrictions

None.

Project License

The software license applied to the tool should be an LGPL or MIT license.

Level of Sponsor Involvement

Teams are free to contact the sponsor for any clarifications through emails or via Microsoft Teams for collaboration. One hour meeting per fortnight will be conducted.

Email: vikrant.kaulgud@accenture.com and kapil.singi@accenture.com