

Øvelse 41 - Code skanning i repositoryet

Information

Formålet med denne øvelse er at introducere konceptet sikkerhedsskanning af kode hos en repository-udbyder, også kaldet `Security code scanner`.

Sikkerhedsskanning er, ligesom *taint-analysis*, en form for statisk kodeanalyse og kan inkludere taint-analyse som en delkomponent.

Forskellen på taint-analyse og kode-skanning

- **Taint-analyse** sporer hvordan data bevæger sig gennem applikationen fra usikre kilder til kritiske operationer (sink). Det er en fokuseret analyse, som hjælper med at finde datavejssårbarheder som SQL Injection eller Cross-site Scripting (XSS).
- **Kode-skanning** (Security code scanning) er bredere og analyserer koden for mange forskellige typer sårbarheder baseret på et stort bibliotek af sikkerhedsregler og mønstre. Den kan identificere et større sæt af problemer såsom usikker konfiguration, sårbar brug af kryptografi, dårlig adgangskontrol, m.m.
Begge er statiske analyser, men med forskelligt fokus og omfang.

Github benytter f.eks. [CodeQL](#) til sikkerhedsskanning, som er en kraftfuld platform for avanceret statisk analyse.

Instruktioner

1. Sæt repoet til `public`

Se vejledning [her](#).

Detter er et krav når man ikke har en betalt udgave af Github (F.eks. Github pro)

2. Aktivér GitHub Code Scanning (CodeQL)

Følg disse trin for at aktivere standardopsætningen for CodeQL-scanning i dit repository:

1. Gå til dit repository på GitHub.
2. Gå derefter til **Settings** → **Security** → **Advanced Security**.
3. Under **Code Security**, find sektionen **CodeQL analysis**, klik på **Set up** → **Default**.
4. Et vindue med **"CodeQL default configuration"** vises.
 - Her kan du se, hvilke sprog GitHub automatisk har registreret.
 - (Valgfrit) Klik **Edit** for at tilpasse:

- Tilføj eller fjern sprog (f.eks. JavaScript, Python, C#).
- Vælg "Query suite" – f.eks. *security-and-quality*.

5. Klik **Enable CodeQL**.

Dette opretter en workflow-fil i `.github/workflows/codeql.yml` og starter den første scanning automatisk.

6. **Vent på, at scannings opsætningen færdiggøres.**

Du kan følge med under fanen **Actions**.

7. **Se resultaterne:**

8. Gå til fanen **Security** → **Code scanning alerts**.

9. Gennemgå fundne sårbarheder.

□ Refleksionsspørgsmål

1. □ Hvordan adskiller sikkerhedsskanning med CodeQL sig fra taint-analyse i praksis?
2. □ Hvilke fordele giver det at have flere lag af statisk analyse i et projekt?
3. □ Hvordan kan man afgøre, hvornår en fundet sårbarhed er en falsk positiv?
4. □ Hvordan kan automatiserede kode-skanninger integreres effektivt i CI/CD pipelines?
5. □ □ □ Hvordan understøtter både taint-analyse og kode-skanning samarbejde og vidensdeling i et udviklingsteam?
6. □ Hvordan bidrager flere typer statisk analyse til en *defense in depth* strategi i softwareudvikling?

□ *At kombinere forskellige statiske analysemetoder, såsom taint-analyse og bredere kode-skanning, styrker evnen til at opdage sårbarheder tidligt og bidrager til en mere sikker softwareudviklingsproces.*

Last update: 2025-10-30 12:07:19