

[GK] 10.11.1 Statische Codeanalyse

@authors: Ertl Jakob, Latschbacher Lukas

Rule 1 - Vorhandensein von Kommentaren bei public Methoden

Diese Regel überprüft ob über öffentlichen Methoden ein Kommentar ist, um diese zu erklären. Falls dies nicht der Fall ist, wird eine Warnung ausgegeben.

```
rules:
- id: kommentare-bei-oeffentlichen-methoden
  patterns:
    - pattern: |
        public $RETURNTYPE $METHOD(...) {
            ...
        }
    - pattern-not: |
        /**
         * ...
         */
        public $RETURNTYPE $METHOD(...) {
            ...
        }
  message: "Öffentliche Methoden sollten Kommentare haben, die ihre
  Funktionalität erklären."
  languages: [java]
  severity: WARNING
```

Rule 2 - Vermeidung von gefaehrlichen Konstrukten

Diese Regel überprüft ob ein gefährliches Konstrukt verwendet wird. In dem Fall haben wir System.exit() gewählt, da diese Methode direkt zum Programmabsturz führt, was nicht unbedingt gewollt ist ohne das gespeichert ist.

```
rules:
- id: vermeidung-von-system-exit
  pattern: System.exit($X);
  message: "Vermeide die Verwendung von System.exit(), um die Applikation
  nicht abrupt zu beenden."
  languages: [java]
  severity: ERROR
```

Ausführen:

```
jacob@jacob-B1NA: ~/ITS1Sengrep$ sengrep scan --config Regel_geregkonstruktiv.yml

Scan Status
Scanning 12 files (only git-tracked) with 1 Code rule:

CODE RULES
Scanning 5 files.

SUPPLY CHAIN RULES
Run 'sengrep ci' to find dependency vulnerabilities and advanced cross-file findings.

PROGRESS
100% 0:00:00

1 Code Finding

Main.java
>>> vermeidung-von-system-exit
Vermeide die Verwendung von System.exit(), um die Applikation nicht abrupt zu beenden.

9: System.exit(x);

Scan Summary
Some files were skipped or only partially analyzed.
Scan was limited to files tracked by git.

Ran 1 rule on 5 files: 1 finding.

A new version of Sengrep is available. See https://sengrep.dev/docs/upgrading
jacob@jacob-B1NA: ~/ITS1Sengrep$ git pull
```

Rule 3 - Erkennung von authentifizierungs Routinen

Diese Regel erkennt Authentifizierungs Routinen und zeigt diese an, damit man die Sicherheit derer überprüfen kann.

```
rules:
- id: authentifizierungs-routinen
  pattern: |
    public $RETURNTYPE login($PARAMS) {
      ...
    }
  message: "Überprüfe die Sicherheit der Authentifizierungsroutine."
  languages: [java]
  severity: INFO
```

Rule 4 - Erkennung von Sicherheitsluecken

Diese Regel erkennt die Sicherheitslücke von hardcodierten Passwörtern.

```
rules:
- id: hardcodierte-variablen-in-login-methode
  patterns:
  - pattern: |
      public $RETURNTYPE login(...) {
        ...
        $VARIABLE = $VALUE;
        ...
      }
```

```
    }  
    message: "Vermeide die Verwendung hartkodierter Variablen innerhalb von  
Login-Methoden."  
    languages: [java]  
    severity: ERROR
```

Ausführen:

```
jakob@jakob-81NA:~/ITSISemgrep$ semgrep scan --config semgrepRules/regel_sicherheitsluecken.yml
```

Scan Status

Scanning 12 files (only git-tracked) with 1 Code rule:

CODE RULES

Scanning 5 files.

SUPPLY CHAIN RULES

💡 Run 'semgrep ci' to find dependency vulnerabilities and advanced cross-file findings.

PROGRESS

100% 0:00:00

1 Code Finding

Main.java

```
>>> semgrepRules.hartcodierte-passwoerter  
Vermeide hartcodierte Passwörter im Code.  
  
16: String $PASSWORD = "passwort";
```

Scan Summary

Some files were skipped or only partially analyzed.
Scan was limited to files tracked by git.

Ran 1 rule on 5 files: 1 finding.

🔔 A new version of Semgrep is available. See <https://semgrep.dev/docs/upgrading>

Rule 5 - unsichere Konfigurationsoptionen

Diese Regel erkennt nicht sicher konfigurierten File Zugriff.

```
rules:  
- id: unsichere-konfiguration  
  pattern: |  
    $X = new FileInputStream("$FILE");  
  message: "Stelle sicher, dass der Zugriff auf Dateien sicher konfiguriert  
ist."  
  languages: [java]  
  severity: WARNING
```

Rule 6 - unsichere Dateipfade

Diese Regel erkennt unsichere Zugriffe auf sicherheitsrelevante Dateipfade, wie etc/passwd.

```
- id: java-unsafe-file-paths
  patterns:
    - pattern: |
        $Path = $any;
        $Pattern.matches(".*etc/passwd", $Path);
      message: "Unsicherer Dateipfad gefunden: möglicher Zugriff auf
      Passwortdateien."
  languages:
    - java
```