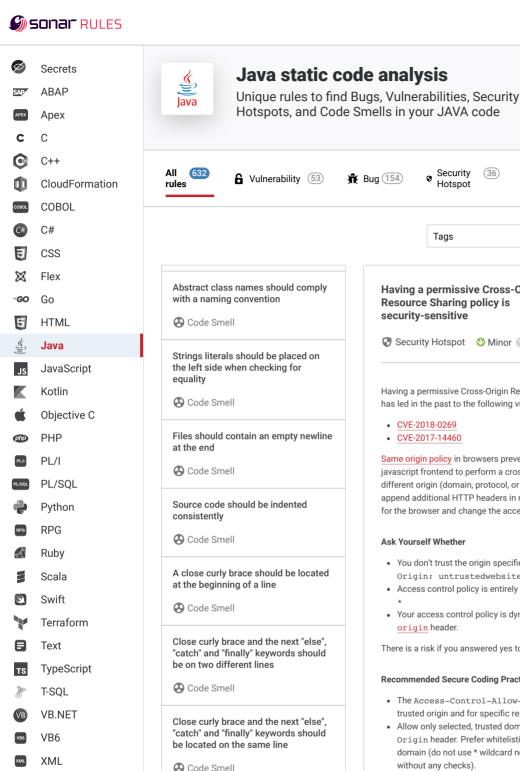
Products ✓

O Quick 42 Fix



at the beginning of a line

Code Smell

at the end of a line

Tabulation characters should not be

Functions should not be defined with a variable number of arguments

Code Smell

Code Smell

Code Smell

used

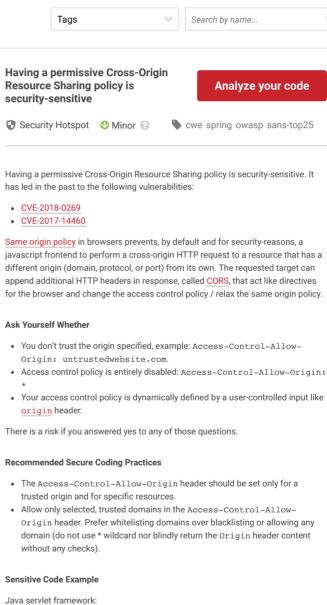


@Override

Spring MVC framework

@CrossOrigin // Sensitive @RequestMapping("")

CrossOrigin



protected void doGet(HttpServletRequest req, HttpServletResp

resp.setHeader("Content-Type", "text/plain; charset=utf-

resp.setHeader("Access-Control-Allow-Origin", "*"); // S resp.setHeader("Access-Control-Allow-Credentials", "true

resp.setHeader("Access-Control-Allow-Methods", "GET");

resp.getWriter().write("response");

⊗ Code

Smell

(389)

(36)

Local-Variable Type Inference should be used

Code Smell

Migrate your tests from JUnit4 to the new JUnit5 annotations

Code Smell

Track uses of disallowed classes

Code Smell

Track uses of "@SuppressWarnings" annotations

Code Smell

```
public class TestController {
    public String home(ModelMap model) {
        model.addAttribute("message", "ok ");
        return "view";
    }
}
```

cors.CorsConfiguration

```
CorsConfiguration config = new CorsConfiguration();
config.addAllowedOrigin("*"); // Sensitive
config.applyPermitDefaultValues(); // Sensitive
```

servlet.config.annotation.CorsConfiguration

```
class Insecure implements WebMvcConfigurer {
  @Override
  public void addCorsMappings(CorsRegistry registry) {
    registry.addMapping("/**")
        .allowedOrigins("*"); // Sensitive
  }
}
```

Compliant Solution

Java Servlet framework:

```
@Override
protected void doGet(HttpServletRequest req, HttpServletResp
    resp.setHeader("Content-Type", "text/plain; charset=utf-
    resp.setHeader("Access-Control-Allow-Origin", "trustedwe
    resp.setHeader("Access-Control-Allow-Credentials", "true
    resp.setHeader("Access-Control-Allow-Methods", "GET");
    resp.getWriter().write("response");
}
```

Spring MVC framework:

CrossOrigin

```
@CrossOrigin("trustedwebsite.com") // Compliant
@RequestMapping("")
public class TestController {
    public String home(ModelMap model) {
        model.addAttribute("message", "ok ");
        return "view";
    }
}
```

cors.CorsConfiguration

```
CorsConfiguration config = new CorsConfiguration();
config.addAllowedOrigin("http://domain2.com"); // Compliant
```

servlet.config.annotation.CorsConfiguration

```
class Safe implements WebMvcConfigurer {
   @Override
   public void addCorsMappings(CorsRegistry registry) {
      registry.addMapping("/**")
      .allowedOrigins("safe.com"); // Compliant
   }
}
```

See

- OWASP Top 10 2021 Category A5 Security Misconfiguration
- OWASP Top 10 2021 Category A7 Identification and Authentication Failures
- developer.mozilla.org CORS
- developer.mozilla.org Same origin policy
- OWASP Top 10 2017 Category A6 Security Misconfiguration
- OWASP HTML5 Security Cheat Sheet Cross Origin Resource Sharing
- MITRE, CWE-346 Origin Validation Error
- <u>MITRE, CWE-942</u> Overly Permissive Cross-domain Whitelist

• SANS Top 25 - Porous Defenses

Available In:

sonarcloud sonarqube

© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOURCE, SONARLINT, SONARQUBE and SONARCLOUD are trademarks of SonarSource S.A. All other trademarks and copyrights are the property of their respective owners. All rights are expressly reserved.

Privacy Policy