SonarQube community

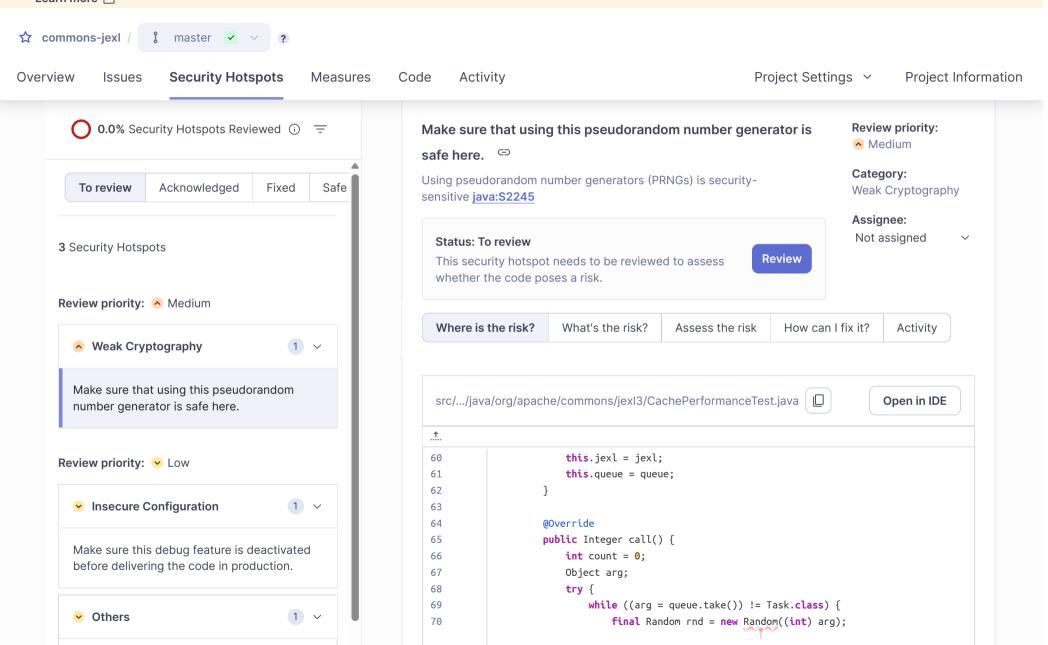Projects   Issues   Rules   Quality Profiles   Quality Gates   Administration   More ⌄

⚠ Embedded database should be used for evaluation purposes only. It doesn't support scaling, upgrading to a new SonarQube Server version, or migration to another database engine.
Learn more ↗

☆ commons-jexl / 🔖 master ✓ ⌄ ❓

Overview   Issues   **Security Hotspots**   Measures   Code   Activity   Project Settings ⌄   Project Information

◯ **0.0%** Security Hotspots Reviewed ⓘ ☰

| To review | Acknowledged | Fixed | Safe |

3 Security Hotspots

Review priority: ⌃ Medium

⌃ **Weak Cryptography**   1 ⌄

> Make sure that using this pseudorandom number generator is safe here.

Review priority: ⌄ Low

⌄ **Insecure Configuration**   1 ⌄

Make sure this debug feature is deactivated before delivering the code in production.

⌄ **Others**   1 ⌄

Make sure that using this pseudorandom number generator is safe here. 🔗

Using pseudorandom number generators (PRNGs) is security-sensitive **java:S2245**

Status: To review
This security hotspot needs to be reviewed to assess whether the code poses a risk.   [Review]

Review priority:
⌃ Medium

Category:
Weak Cryptography

Assignee:
Not assigned ⌄

| Where is the risk? | What's the risk? | Assess the risk | How can I fix it? | Activity |

src/.../java/org/apache/commons/jexl3/CachePerformanceTest.java 📋   [Open in IDE]

⤒

```
60              this.jexl = jexl;
61              this.queue = queue;
62          }
63
64          @Override
65          public Integer call() {
66              int count = 0;
67              Object arg;
68              try {
69                  while ((arg = queue.take()) != Task.class) {
70                      final Random rnd = new Random((int) arg);
```

Make sure publicly writable directories are used safely here.

Make sure that using this pseudorandom number generator is safe here.

```
71                          for (int l = 0; l < LOOPS; ++l) {
72                              for (int c = 0; c < CACHED; ++c) {
73                                  final int ctl = rnd.nextInt(SCRIPTS);
74                                  for (int r = 0; r < HIT; ++r) {
75                                      final JexlScript script = jexl
        .createScript(Integer.toString(ctl));
76                                      final Object result = script.execute(null);
77                                      assertEquals(((Number) result).intValue(), ctl);
78                                      count += 1;
79                                  }
80                              }
```