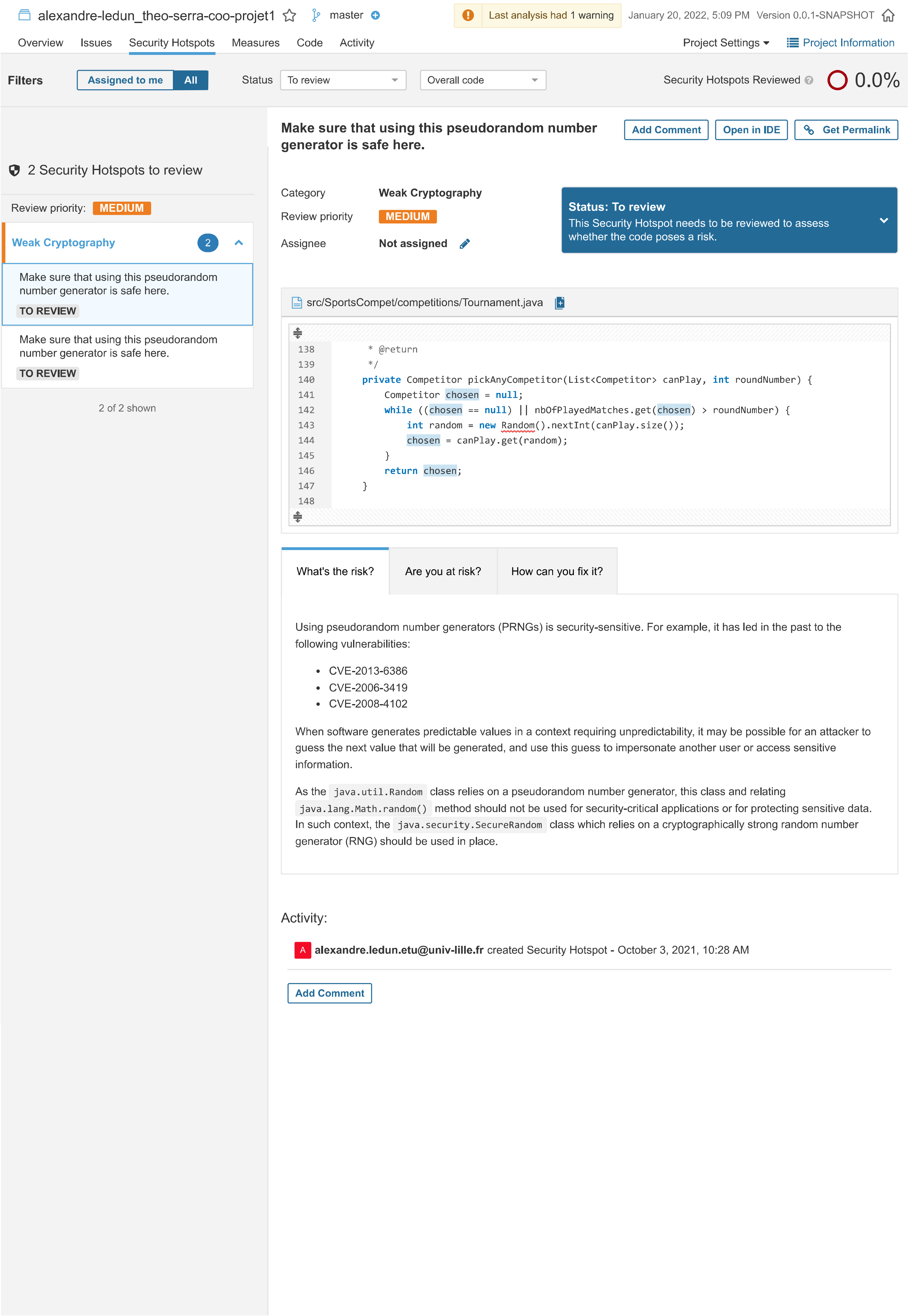

Make sure that using this pseudorandom number generator is safe here.

TO REVIEW

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

TO REVIEW

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

Category **Weak Cryptography**

Review priority **MEDIUM**

Assignee **Not assigned**

Status: To review

This Security Hotspot needs to be reviewed to assess whether the code poses a risk.

src/SportsCompet/competitions/Tournament.java

```
     * @return
     */
    private Competitor pickAnyCompetitor(List<Competitor> canPlay, int roundNumber) {
        Competitor chosen = null;
        while ((chosen == null) || nbOfPlayedMatches.get(chosen) > roundNumber) {
            int random = new Random().nextInt(canPlay.size());
            chosen = canPlay.get(random);
        }
        return chosen;
    }

```

Using pseudorandom number generators (PRNGs) is security-sensitive. For example, it has led in the past to the following vulnerabilities:

- CVE-2013-6386
- CVE-2006-3419
- CVE-2008-4102

When software generates predictable values in a context requiring unpredictability, it may be possible for an attacker to guess the next value that will be generated, and use this guess to impersonate another user or access sensitive information.

As the `java.util.Random` class relies on a pseudorandom number generator, this class and relating `java.lang.Math.random()` method should not be used for security-critical applications or for protecting sensitive data. In such context, the `java.security.SecureRandom` class which relies on a cryptographically strong random number generator (RNG) should be used in place.

Activity:

A **alexandre.ledun.etu@univ-lille.fr** created Security Hotspot - October 3, 2021, 10:28 AM

Add Comment