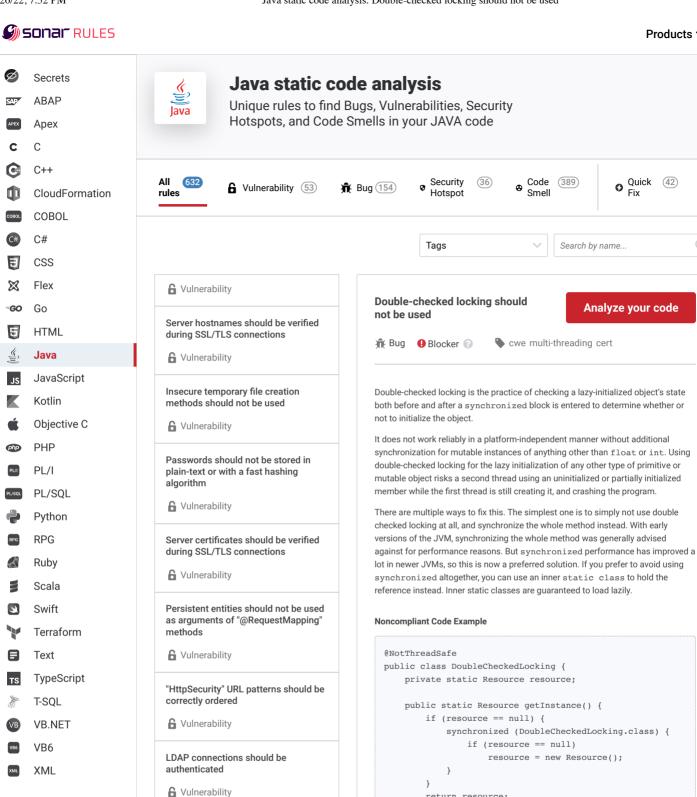
Products >

O Quick 42



Cryptographic keys should be robust

Weak SSL/TLS protocols should not

"SecureRandom" seeds should not be

Cipher Block Chaining IVs should be

Vulnerability

Vulnerability

Vulnerability

unpredictable

Vulnerability

be used

predictable

```
public static Resource getInstance() {
        synchronized (DoubleCheckedLocking.class) {
                resource = new Resource():
    return resource;
static class Resource {
```

Compliant Solution

```
@ThreadSafe
public class SafeLazyInitialization {
   private static Resource resource;
   public static synchronized Resource getInstance() {
        if (resource == null)
           resource = new Resource();
        return resource;
```

Basic authentication should not be

❸ Vulnerability

Regular expressions should not be vulnerable to Denial of Service attacks

■ Vulnerability

"HttpServletRequest.getRequestedSessio should not be used

Vulnerability

Hashes should include an unpredictable salt

♠ Vulnerability

```
static class Resource {
}
```

With inner static holder.

```
@ThreadSafe
public class ResourceFactory {
   private static class ResourceHolder {
       public static Resource resource = new Resource(); //
   public static Resource getResource() {
       return ResourceFactory.ResourceHolder.resource;
    static class Resource {
```

Using "volatile":

```
class ResourceFactory {
 private volatile Resource resource;
  public Resource getResource() {
   Resource localResource = resource;
   if (localResource == null) {
      synchronized (this) {
       localResource = resource;
       if (localResource == null) {
         resource = localResource = new Resource();
    return localResource;
  static class Resource {
}
```

See

- The "Double-Checked Locking is Broken" Declaration
- CERT, LCK10-J. Use a correct form of the double-checked locking idiom
- MITRE, CWE-609 Double-checked locking
- JLS 12.4 Initialization of Classes and Interfaces
- Wikipedia: Double-checked locking

Available In:

sonarlint ⊕ | 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