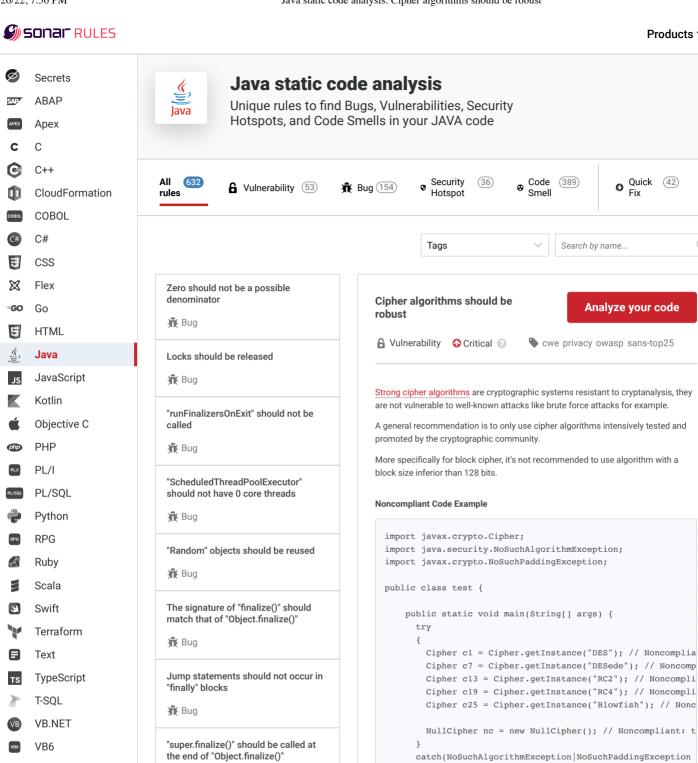
Products ✓

O Quick 42 Fix



## **Compliant Solution**

```
import javax.crypto.Cipher;
import java.security.NoSuchAlgorithmException;
import javax.crypto.NoSuchPaddingException;
public class test {
    public static void main(String[] args) {
      try
        Cipher c31 = Cipher.getInstance("AES/GCM/NoPadding")
      \verb|catch(NoSuchAlgorithmException|| NoSuchPaddingException||\\
```

**XML** 

implementations

security-sensitive

security-sensitive

sensitive

Security Hotspot

Security Hotspot

security-sensitive

Security Hotspot

Using slow regular expressions is

Using publicly writable directories is

Using clear-text protocols is security-

Accessing Android external storage is

₩ Bug

Receiving intents is security-sensitive
Security Hotspot

Broadcasting intents is security-sensitive
Security Hotspot

Expanding archive files without controlling resource consumption is security-sensitive
Security Hotspot

Configuring loggers is security-sensitive

See

• OWASP Top 10 2021 Category A2 - Cryptographic Failures
• OWASP Top 10 2017 Category A3 - Sensitive Data Exposure
• Mobile AppSec Verification Standard - Cryptography Requirements
• OWASP Mobile Top 10 2016 Category M5 - Insufficient Cryptography
• MITRE, CWE-327 - Use of a Broken or Risky Cryptographic Algorithm
• CERT, MSC61-J. - Do not use insecure or weak cryptographic algorithms
• SANS Top 25 - Porous Defenses

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