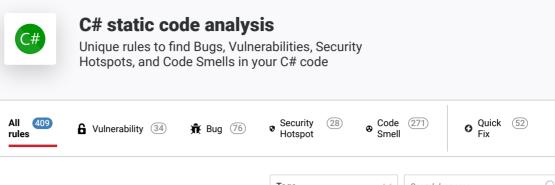
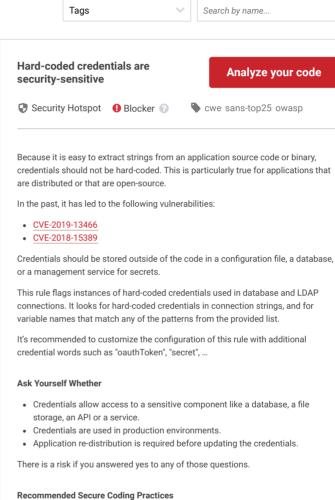


Bua

expected fields

Getters and setters should access the





- Store the credentials in a configuration file that is not pushed to the code repository.
- Store the credentials in a database.
- Use your cloud provider's service for managing secrets.
- If a password has been disclosed through the source code: change it.

Sensitive Code Example

```
string username = "admin";
string password = "Admin123"; // Sensitive
string usernamePassword = "user=admin&password=Admin123"; /
string url = "scheme://user:Admin123@domain.com"; // Sensiti
```

Compliant Solution

```
string username = "admin";
string password = GetEncryptedPassword();
string usernamePassword = string.Format("user={0}&password={
    string url = $"scheme://{username}:{password}@domain.com";
```



Right operands of shift operators should be integers



Shared resources should not be used for locking



Locks should be released



Using publicly writable directories is security-sensitive



string url2 = "http://guest:guest@domain.com"; // Compliant
const string Password_Property = "custom.password"; // Compl

Exceptions

- Issue is not raised when URI username and password are the same.
- Issue is not raised when searched pattern is found in variable name and value.

See

- OWASP Top 10 2021 Category A7 Identification and Authentication Failures
- OWASP Top 10 2017 Category A2 Broken Authentication
- MITRE, CWE-798 Use of Hard-coded Credentials
- MITRE, CWE-259 Use of Hard-coded Password
- SANS Top 25 Porous Defenses
- Derived from FindSecBugs rule Hard Coded Password

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