

⊌ HTML

Java
Js JavaScript

Kotlin

Kubernetes

Objective CPHP

PL/I

PL/SQL PL/SQL

Python

RPG RPG

Ruby

Scala

Swift

Terraform

■ Text

Ts TypeScript

T-SQL

VB VB.NET

VB6 VB6

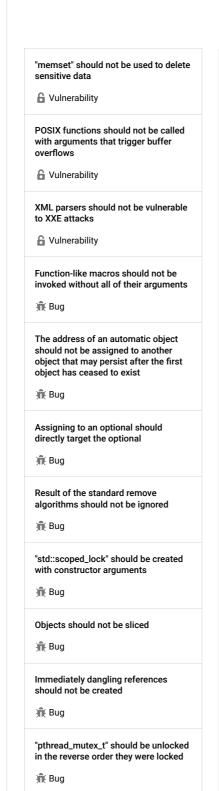
xmL XML



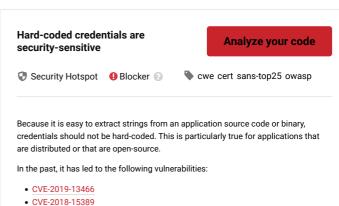
Unique rules to find Bugs, Vulnerabilities, Security Hotspots, and Code Smells in your C++ code



Tags



"pthread_mutex_t" should be properly



Search by name.

Credentials should be stored outside of the code in a configuration file, a database, or a management service for secrets.

This rule looks for hard-coded credentials in variable names that match any of the patterns from the provided list.

Ask Yourself Whether

- Credentials allow access to a sensitive component like a database, a file storage, an API or a service.
- · Credentials are used in production environments.
- Application re-distribution is required before updating the credentials.

There is a risk if you answered yes to any of those questions.

Recommended Secure Coding Practices

- 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

```
dbi_conn conn = dbi_conn_new("mysql");
string password = "secret"; // Sensitive
dbi_conn_set_option(conn, "password", password.c_str());
```

Compliant Solution

```
dbi_conn conn = dbi_conn_new("mysql");
string password = getDatabasePassword(); // Compliant
dbi_conn_set_option(conn, "password", password.c_str()); // C
```

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
 CERT, MSC03-J. Never hard code sensitive information

initialized and destroyed

Bug

"pthread_mutex_t" should not be consecutively locked or unlocked twice

Bug

"std::move" and "std::forward" should not be confused

Bug

A call to "wait()" on a "std::condition_variable" should have a

SANS Top 25 - Porous Defenses
 Derived from FindSecBugs rule Hard Coded Password

Available In:
sonarcloud Sonarqube
Developer Edition

© 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