C# static code analysis

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

Hard-coded credentials are security-sensitive

Analyze your code

Security Hotspot   Blocker   cwe sans-top25 owasp

Because it is easy to extract strings from an application source code or binary, credentials should not be hard-coded. This is particularly true for applications that are distributed or that are open-source.

In the past, it has led to the following vulnerabilities:

- CVE-2019-13466
- CVE-2018-15389

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

This rule flags instances of hard-coded credentials used in database and LDAP connections. It looks for hard-coded credentials in connection strings, and for variable names that match any of the patterns from the provided list.

It's recommended to customize the configuration of this rule with additional credential words such as "oauthToken", "secret", ...

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

```
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";
```

Bug

Right operands of shift operators should be integers

Bug

Shared resources should not be used for locking

Bug

Locks should be released

Bug

Using publicly writable directories is security-sensitive

Security Hotspot

```
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