- Secrets
- ABAP
- Apex
- C
- C++
- CloudFormation
- COBOL
- C#
- CSS
- Flex
- Go
- HTML
- Java
- JavaScript
- Kotlin
- Kubernetes
- Objective C
- PHP
- PL/I
- PL/SQL
- Python
- RPG
- Ruby
- Scala
- Swift
- Terraform
- Text
- TypeScript
- T-SQL
- VB.NET
- VB6
- XML

T-SQL static code analysis

Unique rules to find Bugs, Vulnerabilities, Security Hotspots, and Code Smells in your T-SQL code

All rules 80 | Vulnerability 1 | Bug 16 | Security Hotspot 4 | Code Smell 59

Tags

Search by name...

Hard-coded credentials are security-sensitive
Security Hotspot

The number of variables in a FETCH statement should match the number of columns in the cursor
Bug

"CASE" input expressions should be invariant
Bug

Nullable subqueries should not be used in "NOT IN" conditions
Bug

Using weak hashing algorithms is security-sensitive
Security Hotspot

Dynamically executing code is security-sensitive
Security Hotspot

"SELECT" statements used as argument of "EXISTS" statements should be selective
Code Smell

Size should be specified for "varchar" variables and parameters
Code Smell

Conditionally executed code should be denoted by either indentation or BEGIN...END block
Code Smell

Conditionals should start on new lines
Code Smell

"INSERT" statements should explicitly list the columns to be set
Code Smell

Output parameters should be assigned

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 allows 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 the a password has been disclosed through the source code: change it.

See

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

Available In:
sonarcloud sonarqube Developer Edition