





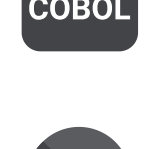











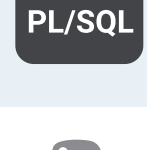


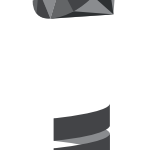













-  Secrets
-  ABAP
-  Apex
-  C
-  C++
-  CloudFormation
-  COBOL
-  C#
-  CSS
-  Flex
-  Go
-  HTML
-  Java
-  JavaScript
-  Kotlin
-  Kubernetes
-  Objective C
-  PHP
-  PL/I
-  **PL/SQL**
-  Python
-  RPG
-  Ruby
-  Scala
-  Swift
-  Terraform
-  Text
-  TypeScript
-  T-SQL
-  VB.NET
-  VB6
-  XML


PL/SQL













PL/SQL static code analysis

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

- All rules 188
-  Vulnerability 4
-  Bug 45
-  Security Hotspot 2
-  Code Smell 137

Tags 

Search by name... 

 Code Smell
Neither DES (Data Encryption Standard) nor DESede (3DES) should be used
 Vulnerability
"SYNCHRONIZE" should not be used
 Bug
Global public variables should not be defined
 Code Smell
A primary key should be specified during table creation
 Code Smell
Track lack of copyright and license headers
 Code Smell
SHA-1 and Message-Digest hash algorithms should not be used in secure contexts
 Vulnerability
Sensitive "SYS" owned functions should not be used
 Vulnerability
"FORMS_DDL('COMMIT')" and "FORMS_DDL('ROLLBACK')" should not be used
 Bug
"DBMS_OUTPUT.PUT_LINE" should not be used
 Code Smell
"WHEN OTHERS" should not be the only exception handler
 Code Smell
"INSERT" statements should explicitly list the columns to be set
 Code Smell

Neither DES (Data Encryption Standard) nor DESede (3DES) should be used

Analyze your code

-  Vulnerability
-  Blocker 
-  cwe owasp sans-top25

According to the US National Institute of Standards and Technology (NIST), the Data Encryption Standard (DES) is no longer considered secure:

Adopted in 1977 for federal agencies to use in protecting sensitive, unclassified information, the DES is being withdrawn because it no longer provides the security that is needed to protect federal government information. Federal agencies are encouraged to use the Advanced Encryption Standard, a faster and stronger algorithm approved as FIPS 197 in 2001.

For similar reasons, RC2 should also be avoided.

Noncompliant Code Example

```
PLS_INTEGER := DBMS_CRYPTO.ENCRYPT_DES
               + DBMS_CRYPTO.CHAIN_CBC
               + DBMS_CRYPTO.PAD_PKCS5;
```

Compliant Solution

```
PLS_INTEGER := DBMS_CRYPTO.ENCRYPT_AES256
               + DBMS_CRYPTO.CHAIN_CBC
               + DBMS_CRYPTO.PAD_PKCS5;
```

See

- OWASP Top 10 2017 Category A6 - Security Misconfiguration
- MITRE, CWE-326 - Inadequate Encryption Strength
- 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
- Derived from FindSecBugs rule [DES / DESede Unsafe](#)

Deprecated

This rule is deprecated; use {rule:plsqli:S5547} instead.

Available In:

 |  |  Developer Edition