
















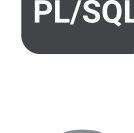
















-  Secrets
-  ABAP
-  Apex
-  C
-  C++
-  CloudFormation
-  COBOL
-  C#
-  CSS
-  Flex
-  Go
-  HTML
-  Java
-  JavaScript
-  Kotlin
-  Kubernetes
-  Objective C
-  PHP
-  PL/I
-  PL/SQL
-  Python
-  RPG
-  Ruby
-  Scala
-  Swift
-  Terraform
-  Text
-  TypeScript
-  **T-SQL**
-  VB.NET
-  VB6
-  XML














# 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... 🔍

Using hardcoded IP addresses is security-sensitive	 Security Hotspot
Column references should not have more than two-parts	 Code Smell
Triggers should not "PRINT", "SELECT", or "FETCH"	 Code Smell
"LIKE" clauses should not be used without wildcards	 Code Smell
Jump statements should not be redundant	 Code Smell
"CATCH" clauses should do more than rethrow	 Code Smell
Boolean checks should not be inverted	 Code Smell
Multiple variables should not be declared on the same line	 Code Smell
Unused local variables should be removed	 Code Smell
Local variable and parameter names should comply with a naming convention	 Code Smell
Empty statements should be removed	 Code Smell

## Using hardcoded IP addresses is security-sensitive

[Analyze your code](#)

 Security Hotspot  Minor   owasp

Hardcoding IP addresses is security-sensitive. It has led in the past to the following vulnerabilities:

- [CVE-2006-5901](#)
- [CVE-2005-3725](#)

Today's services have an ever-changing architecture due to their scaling and redundancy needs. It is a mistake to think that a service will always have the same IP address. When it does change, the hardcoded IP will have to be modified too. This will have an impact on the product development, delivery and deployment:

- The developers will have to do a rapid fix every time this happens, instead of having an operation team change a configuration file.
- It forces the same address to be used in every environment (dev, sys, qa, prod).

Last but not least it has an effect on application security. Attackers might be able to decompile the code and thereby discover a potentially sensitive address. They can perform a Denial of Service attack on the service at this address or spoof the IP address. Such an attack is always possible, but in the case of a hardcoded IP address the fix will be much slower, which will increase an attack's impact.

### Ask Yourself Whether

The disclosed IP address is sensitive, eg:

- Can give information to an attacker about the network topology.
- It's a personal (assigned to an identifiable person) IP address.

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

### Recommended Secure Coding Practices

Don't hard-code the IP address in the source code, instead make it configurable.

### Sensitive Code Example

```
SET @IP = '192.168.12.42'; -- Sensitive
```

### Compliant Solution

```
SET @IP = (SELECT ip_address FROM configuration); -- Compliant
```

### Exceptions

No issue is reported for the following cases because they are not considered sensitive:

- Loopback addresses 127.0.0.0/8 in CIDR notation (from 127.0.0.0 to 127.255.255.255)
- Broadcast address 255.255.255.255
- Non routable address 0.0.0.0
- Strings of the form 2.5.<number>.<number> as they **often match Object Identifiers** (OID).

### See

- [OWASP Top 10 2017 Category A3](#) - Sensitive Data Exposure
- [CERT, MSC03-J](#) - Never hard code sensitive information

Available In:

**sonarcloud**  | **sonarqube**  Developer Edition