CWE Common Weakness Enumeration
A community-developed list of SW & HW weaknesses that can become vulnerabilities

Top 25 | Top HW CWE | New to CWE? Start here!

Home > CWE List > CWE- Individual Dictionary Definition (4.15) ID Lookup: Go

Home | About ▼ | CWE List ▼ | Mapping ▼ | Top-N Lists ▼ | Community ▼ | News ▼ | Search

CWE-546: Suspicious Comment

Weakness ID: 546
Vulnerability Mapping: ALLOWED
Abstraction: Variant

View customized information: Conceptual | Operational | Mapping Friendly | **Complete** | Custom

Description

The code contains comments that suggest the presence of bugs, incomplete functionality, or weaknesses.

Extended Description

Many suspicious comments, such as BUG, HACK, FIXME, LATER, LATER2, TODO, in the code indicate missing security functionality and checking. Others indicate code problems that programmers should fix, such as hard-coded variables, error handling, not using stored procedures, and performance issues.

Common Consequences

Scope	Impact	Likelihood
Other	**Technical Impact:** *Quality Degradation* Suspicious comments could be an indication that there are problems in the source code that may need to be fixed and is an indication of poor quality. This could lead to further bugs and the introduction of weaknesses.	

Potential Mitigations

Phase: Documentation

Remove comments that suggest the presence of bugs, incomplete functionality, or weaknesses, before deploying the application.

Relationships

Relevant to the view "Research Concepts" (CWE-1000)

Nature	Type	ID	Name
ChildOf	C	1078	Inappropriate Source Code Style or Formatting
PeerOf	V	615	Inclusion of Sensitive Information in Source Code Comments

Modes Of Introduction

Phase	Note
Implementation	

Applicable Platforms

Languages

Class: Not Language-Specific *(Undetermined Prevalence)*

Demonstrative Examples

Example 1

The following excerpt demonstrates the use of a suspicious comment in an incomplete code block that may have security repercussions.

Example Language: **Java** *(bad code)*

```java
if (user == null) {
  // TODO: Handle null user condition.
}
```

Weakness Ordinalities

Ordinality	Description
Indirect	*(where the weakness is a quality issue that might indirectly make it easier to introduce security-relevant weaknesses or make them more difficult to detect)*

Memberships

Nature	Type	ID	Name
MemberOf	V	884	CWE Cross-section
MemberOf	C	963	SFP Secondary Cluster: Exposed Data
MemberOf	C	1412	Comprehensive Categorization: Poor Coding Practices

Vulnerability Mapping Notes

Usage: ALLOWED *(this CWE ID could be used to map to real-world vulnerabilities)*

Reason: Acceptable-Use

Rationale:

This CWE entry is at the Variant level of abstraction, which is a preferred level of abstraction for mapping to the root causes of vulnerabilities.

Comments:

Carefully read both the name and description to ensure that this mapping is an appropriate fit. Do not try to 'force' a mapping to a lower-level Base/Variant simply to comply with this preferred level of abstraction.

Content History

Submissions

Submission Date	Submitter	Organization
2006-07-19 *(CWE Draft 3, 2006-07-19)*	Anonymous Tool Vendor (under NDA)	

Modifications

Page Last Updated: July 16, 2024

Site Map | Terms of Use | Manage Cookies | Cookie Notice | Privacy Policy | Contact Us |