



A Community-Developed List of Software Weakness Types


[Home](#)
[About](#)
[CWE List](#)
[Scoring
Search](#)
[Community](#)
[News](#)
Weakness ID: 359
Abstraction: Class

Structure: Simple

Status: Incomplete

Presentation Filter: Basic

▼ Description

The software does not properly prevent private data (such as credit card numbers) from being accessed by actors who either (1) are not explicitly authorized to access the data or (2) do not have the implicit consent of the people to which the data is related.

▼ Extended Description

Mishandling private information, such as customer passwords or Social Security numbers, can compromise user privacy and is often illegal. An exposure of private information does not necessarily prevent the software from working properly, and in fact it might be intended by the developer, but it can still be undesirable (or explicitly prohibited by law) for the people who are associated with this private information.

Privacy violations may occur when:

1. Private user information enters the program.
2. The data is written to an external location, such as the console, file system, or network.

Private data can enter a program in a variety of ways:

1. Directly from the user in the form of a password or personal information
2. Accessed from a database or other data store by the application
3. Indirectly from a partner or other third party

Some types of private information include:

- Government identifiers, such as Social Security Numbers
- Contact information, such as home addresses and telephone numbers
- Geographic location - where the user is (or was)
- Employment history
- Financial data - such as credit card numbers, salary, bank accounts, and debts
- Pictures, video, or audio
- Behavioral patterns - such as web surfing history, when certain activities are performed, etc.
- Relationships (and types of relationships) with others - family, friends, contacts, etc.
- Communications - e-mail addresses, private e-mail messages, SMS text messages, chat logs, etc.
- Health - medical conditions, insurance status, prescription records
- Credentials, such as passwords, which can be used to access other information.

Some of this information may be characterized as PII (Personally Identifiable Information), Protected Health Information (PHI), etc. Categories of private information may overlap or vary based on the intended usage or the policies and practices of a particular industry.

Depending on its location, the type of business it conducts, and the nature of any private data it handles, an organization may be required to comply with one or more of the following federal and state regulations: - Safe Harbor Privacy Framework [[REF-340](#)] - Gramm-Leach Bliley Act (GLBA) [[REF-341](#)] - Health Insurance Portability and Accountability Act (HIPAA) [[REF-342](#)] - California SB-1386 [[REF-343](#)].

WIZ, Inc. EXHIBIT - 1059

WIZ, Inc. v. Orca Security LTD.

Sometimes data that is not labeled as private can have a privacy implication in a different context. For example, student identification numbers are usually not considered private because there is no explicit and publicly-available mapping to an individual student's personal information. However, if a school generates identification numbers based on student social security numbers, then the identification numbers should be considered private.

Security and privacy concerns often seem to compete with each other. From a security perspective, all important operations should be recorded so that any anomalous activity can later be identified. However, when private data is involved, this practice can in fact create risk. Although there are many ways in which private data can be handled unsafely, a common risk stems from misplaced trust. Programmers often trust the operating environment in which a program runs, and therefore believe that it is acceptable store private information on the file system, in the registry, or in other locally-controlled resources. However, even if access to certain resources is restricted, this does not guarantee that the individuals who do have access can be trusted.

▼ Relationships

The table(s) below shows the weaknesses and high level categories that are related to this weakness. These relationships are defined as ChildOf, ParentOf, MemberOf and give insight to similar items that may exist at higher and lower levels of abstraction. In addition, relationships such as PeerOf and CanAlsoBe are defined to show similar weaknesses that the user may want to explore.

- ▶ **Relevant to the view "Research Concepts" (CWE-1000)**
- ▶ **Relevant to the view "Architectural Concepts" (CWE-1008)**
- ▶ **Relevant to the view "Development Concepts" (CWE-699)**

▼ Modes Of Introduction

The different Modes of Introduction provide information about how and when this weakness may be introduced. The Phase identifies a point in the software life cycle at which introduction may occur, while the Note provides a typical scenario related to introduction during the given phase.

Phase	Note
Architecture and Design	OMISSION: This weakness is caused by missing a security tactic during the architecture and design phase.
Implementation	
Operation	

▼ Applicable Platforms

The listings below show possible areas for which the given weakness could appear. These may be for specific named Languages, Operating Systems, Architectures, Paradigms, Technologies, or a class of such platforms. The platform is listed along with how frequently the given weakness appears for that instance.

Languages

Class: Language-Independent (*Undetermined Prevalence*)

Paradigms

Mobile (*Undetermined Prevalence*)

▼ Common Consequences

The table below specifies different individual consequences associated with the weakness. The Scope identifies the application security area that is violated, while the Impact describes the negative technical impact that arises if an adversary succeeds in exploiting this weakness. The Likelihood provides information about how likely the specific consequence is expected to be seen relative to the other consequences in the list. For example, there may be high likelihood that a weakness will be exploited to achieve a certain impact, but a low likelihood that it will be exploited to achieve a different impact.

Scope	Impact	Likelihood
Confidentiality	Technical Impact: Read Application Data	

▼ Demonstrative Examples

Example 1

In 2004, an employee at AOL sold approximately 92 million private customer e-mail addresses to a spammer marketing an offshore gambling web site [REF-338]. In response to such high-profile exploits, the collection and management of private data is becoming increasingly regulated.

Example 2

The following code contains a logging statement that tracks the contents of records added to a database by storing them in a log file. Among other values that are stored, the getPassword() function returns the user-supplied plaintext password associated with the account.

The code in the example above logs a plaintext password to the filesystem. Although many developers trust the filesystem as a safe storage location for data, it should not be trusted implicitly, particularly when privacy is a concern.

Example 3

This code uses location to determine the user's current US State location.

First the application must declare that it requires the ACCESS_FINE_LOCATION permission in the application's manifest.xml:

During execution, a call to getLastLocation() will return a location based on the application's location permissions. In this case the application has permission for the most accurate location possible:

While the application needs this information, it does not need to use the ACCESS_FINE_LOCATION permission, as the ACCESS_COARSE_LOCATION permission will be sufficient to identify which US state the user is in.

▼ Memberships

This MemberOf Relationships table shows additional CWE Categories and Views that reference this weakness as a member. This information is often useful in understanding where a weakness fits within the context of external information sources.

Nature	Type	ID	Name
MemberOf		254	7PK - Security Features
MemberOf		857	The CERT Oracle Secure Coding Standard for Java (2011). Chapter 14 - Input Output (FIO).
MemberOf		975	SFP Secondary Cluster: Architecture
MemberOf		1029	OWASP Top Ten 2017 Category A3 - Sensitive Data Exposure
MemberOf		1147	SEI CERT Oracle Secure Coding Standard for Java - Guidelines 13. Input Output (FIO).

More information is available — Please select a different filter.

Use of the Common Weakness Enumeration and the associated references from this website are subject to the [Terms of Use](#). For more information, please email cwe@mitre.org.

CWE is sponsored by [US-CERT](#) in the office of [Cybersecurity and Communications](#) at the [U.S. Department of Homeland Security](#). Copyright © 2006-2019, The MITRE Corporation. CWE, CWSS, CWRAF, and the CWE logo are trademarks of [The MITRE Corporation](#).

[Privacy Policy](#)
[Terms of Use](#)
[Site Map](#)
[Contact Us](#)