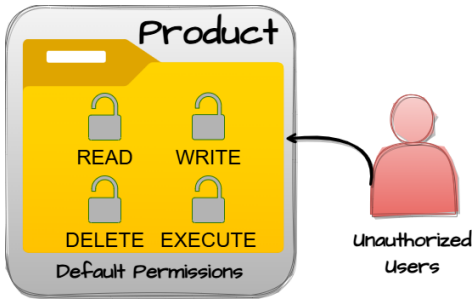
CWE-276: Incorrect Default Permissions

Weakness ID: 276
Vulnerability Mapping: ALLOWED
Abstraction: Base

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

Description

During installation, installed file permissions are set to allow anyone to modify those files.



Common Consequences

Impact	Details
Read Application Data; Modify Application Data	**Scope: Confidentiality, Integrity**

Potential Mitigations

Phase(s)	Mitigation
Architecture and Design; Operation	The architecture needs to access and modification attributes for files to only those users who actually require those actions.
Architecture and Design	**Strategy: *Separation of Privilege*** Compartmentalize the system to have "safe" areas where trust boundaries can be unambiguously drawn. Do not allow sensitive data to go outside of the trust boundary and always be careful when interfacing with a compartment outside of the safe area. Ensure that appropriate compartmentalization is built into the system design, and the compartmentalization allows for and reinforces privilege separation functionality. Architects and designers should rely on the principle of least privilege to decide the appropriate time to use privileges and the time to drop privileges.

Relationships

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

Nature	Type	ID	Name
ChildOf	C	732	Incorrect Permission Assignment for Critical Resource

Relevant to the view "Software Development" (View-699)

Nature	Type	ID	Name
MemberOf	C	275	Permission Issues

Relevant to the view "Weaknesses for Simplified Mapping of Published Vulnerabilities" (View-1003)

Nature	Type	ID	Name
ChildOf	C	732	Incorrect Permission Assignment for Critical Resource

Relevant to the view "Architectural Concepts" (View-1008)

Nature	Type	ID	Name
MemberOf	C	1011	Authorize Actors

Relevant to the view "Hardware Design" (View-1194)

Nature	Type	ID	Name
MemberOf	C	1198	Privilege Separation and Access Control Issues

Modes Of Introduction

Phase	Note
Architecture and Design	
Implementation	
Installation	
Operation	

Applicable Platforms

Languages	Class: Not Language-Specific *(Undetermined Prevalence)*
Technologies	Class: Not Technology-Specific *(Undetermined Prevalence)* Class: ICS/OT *(Undetermined Prevalence)*

Likelihood Of Exploit

Medium

Selected Observed Examples

Note: this is a curated list of examples for users to understand the variety of ways in which this weakness can be introduced. It is not a complete list of all CVEs that are related to this CWE entry.

Reference	Description
CVE-2005-1941	Executables installed world-writable.
CVE-2002-1713	Home directories installed world-readable.
CVE-2001-1550	World-writable log files allow information loss; world-readable file has cleartext passwords.
CVE-2002-1711	World-readable directory.
CVE-2002-1844	Windows product uses insecure permissions when installing on Solaris (genesis: port error).
CVE-2001-0497	Insecure permissions for a shared secret key file. Overlaps cryptographic problem.
CVE-1999-0426	Default permissions of a device allow IP spoofing.

Weakness Ordinalities

Ordinality	Description
Primary	*(where the weakness exists independent of other weaknesses)*

Detection Methods

Method	Details
Automated Static Analysis - Binary or Bytecode	According to SOAR [REF-1479], the following detection techniques may be useful: Cost effective for partial coverage: • Inter-application Flow Analysis **Effectiveness: SOAR Partial**

Manual Static Analysis - Binary or Bytecode	According to SOAR [REF-1479], the following detection techniques may be useful: Cost effective for partial coverage: • Binary / Bytecode disassembler - then use manual analysis for vulnerabilities & anomalies **Effectiveness: SOAR Partial**
Dynamic Analysis with Automated Results Interpretation	According to SOAR [REF-1479], the following detection techniques may be useful: Cost effective for partial coverage: • Host-based Vulnerability Scanners - Examine configuration for flaws, verifying that audit mechanisms work, ensure host configuration meets certain predefined criteria • Web Application Scanner • Web Services Scanner • Database Scanners **Effectiveness: SOAR Partial**
Dynamic Analysis with Manual Results Interpretation	According to SOAR [REF-1479], the following detection techniques may be useful: Highly cost effective: • Host Application Interface Scanner Cost effective for partial coverage: • Fuzz Tester • Framework-based Fuzzer • Automated Monitored Execution • Forced Path Execution **Effectiveness: High**
Manual Static Analysis - Source Code	According to SOAR [REF-1479], the following detection techniques may be useful: Highly cost effective: • Manual Source Code Review (not inspections) Cost effective for partial coverage: • Focused Manual Spotcheck - Focused manual analysis of source **Effectiveness: High**
Automated Static Analysis - Source Code	According to SOAR [REF-1479], the following detection techniques may be useful: Cost effective for partial coverage: • Context-configured Source Code Weakness Analyzer **Effectiveness: SOAR Partial**
Automated Static Analysis	According to SOAR [REF-1479], the following detection techniques may be useful: Cost effective for partial coverage: • Configuration Checker **Effectiveness: SOAR Partial**
Architecture or Design Review	According to SOAR [REF-1479], the following detection techniques may be useful: Highly cost effective: • Formal Methods / Correct-By-Construction

Cost effective for partial coverage:

- Inspection (IEEE 1028 standard) (can apply to requirements, design, source code, etc.)

Effectiveness: High

Memberships

Nature	Type	ID	Name
MemberOf	C	743	CERT C Secure Coding Standard (2008) Chapter 10 - Input Output (FIO)
MemberOf	C	857	The CERT Oracle Secure Coding Standard for Java (2011) Chapter 14 - Input Output (FIO)
MemberOf	C	877	CERT C++ Secure Coding Section 09 - Input Output (FIO)
MemberOf	C	946	SFP Secondary Cluster: Insecure Resource Permissions
MemberOf	C	1147	SEI CERT Oracle Secure Coding Standard for Java - Guidelines 13. Input Output (FIO)
MemberOf	V	1337	Weaknesses in the 2021 CWE Top 25 Most Dangerous Software Weaknesses
MemberOf	C	1345	OWASP Top Ten 2021 Category A01:2021 - Broken Access Control
MemberOf	C	1366	ICS Communications: Frail Security in Protocols
MemberOf	C	1376	ICS Engineering (Construction/Deployment): Security Gaps in Commissioning
MemberOf	V	1387	Weaknesses in the 2022 CWE Top 25 Most Dangerous Software Weaknesses
MemberOf	C	1396	Comprehensive Categorization: Access Control
MemberOf	V	1425	Weaknesses in the 2023 CWE Top 25 Most Dangerous Software Weaknesses
MemberOf	C	1436	OWASP Top Ten 2025 Category A01:2025 - Broken Access Control

Vulnerability Mapping Notes

Usage	**ALLOWED** *(this CWE ID may be used to map to real-world vulnerabilities)*
Reason	Acceptable-Use
Rationale	This CWE entry is at the Base 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.

Notes

Maintenance

As of CWE 4.19, this entry is being considered for deprecation or significant revision. Its name and description are inconsistent. The name is more general, and the description is more specific. The description emphasizes the installation phase only; mentions only files; and emphasizes modification of those files. The name applies to any type of resource, does not mention the specific permissions, and could be relevant to any SDLC phase.

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
PLOVER			Insecure Default Permissions
CERT C Secure Coding	FIO06-C		Create files with appropriate access permissions
The CERT Oracle Secure Coding Standard for Java (2011)	FIO01-J		Create files with appropriate access permission
ISA/IEC 62443	Part 2-4		Req SP.03.08
ISA/IEC 62443	Part 4-2		Req CR 2.1

Related Attack Patterns

CAPEC-ID	Attack Pattern Name
CAPEC-1	Accessing Functionality Not Properly Constrained by ACLs
CAPEC-127	Directory Indexing
CAPEC-81	Web Server Logs Tampering

References

[REF-62]	Mark Dowd, John McDonald and Justin Schuh. *"The Art of Software Security Assessment".* Chapter 3, "Insecure Defaults", Page 69. 1st Edition. Addison Wesley. 2006.
[REF-1479]	Gregory Larsen, E. Kenneth Hong Fong, David A. Wheeler and Rama S. Moorthy. *"State-of-the-Art Resources (SOAR) for Software Vulnerability Detection, Test, and Evaluation".* 2014-07. <https://www.ida.org/-/media/feature/publications/s/st/stateoftheart-resources-soar-for-software-vulnerability-detection-test-and-evaluation/p-5061.ashx>. (*URL validated: 2025-09-05*)
[REF-1479]	Gregory Larsen, E. Kenneth Hong Fong, David A. Wheeler and Rama S. Moorthy. *"State-of-the-Art Resources (SOAR) for Software Vulnerability Detection, Test, and Evaluation".* 2014-07. <https://www.ida.org/-/media/feature/publications/s/st/stateoftheart-resources-soar-for-software-vulnerability-detection-test-and-evaluation/p-5061.ashx>. (*URL validated: 2025-09-05*)

Content History

Submissions

Submission Date	Submitter	Organization
2006-07-19 *(CWE Draft 3, 2006-07-19)*	PLOVER	

Contributions

Contribution Date	Contributor	Organization
2023-04-25	"Mapping CWE to 62443" Sub-Working Group *Suggested mappings to ISA/IEC 62443.*	CWE-CAPEC ICS/OT SIG
2024-09-29 *(CWE 4.19, 2025-12-11)*	Abhi Balakrishnan *Contributed usability diagram concepts used by the CWE team*	

Modifications

Previous Entry Names