# CWE Detail – CWE-552

## Description

The product makes files or directories accessible to unauthorized actors, even though they should not be.

## Extended Description

Web servers, FTP servers, and similar servers may store a set of files underneath a "root" directory that is accessible to the server's users. Applications may store sensitive files underneath this root without also using access control to limit which users may request those files, if any. Alternately, an application might package multiple files or directories into an archive file (e.g., ZIP or tar), but the application might not exclude sensitive files that are underneath those directories. In cloud technologies and containers, this weakness might present itself in the form of misconfigured storage accounts that can be read or written by a public or anonymous user.

## Threat-Mapped Scoring

Score: 0.0

Priority: Unclassified

## Observed Examples (CVEs)

**•** CVE-2005-1835: Data file under web root.

## Related Attack Patterns (CAPEC)

* CAPEC-150
* CAPEC-639

## Attack TTPs

**•** T1003: OS Credential Dumping (Tactics: credential-access)

**•** T1602: Data from Configuration Repository (Tactics: collection)

**•** T1119: Automated Collection (Tactics: collection)

**•** T1530: Data from Cloud Storage (Tactics: collection)

**•** T1555: Credentials from Password Stores (Tactics: credential-access)

**•** T1552.004: Private Keys (Tactics: credential-access)

**•** T1552.003: Bash History (Tactics: credential-access)

**•** T1552.001: Credentials In Files (Tactics: credential-access)

**•** T1552.006: Group Policy Preferences (Tactics: credential-access)

**•** T1039: Data from Network Shared Drive (Tactics: collection)

**•** T1213: Data from Information Repositories (Tactics: collection)

## Modes of Introduction

**•** Architecture and Design: N/A

**•** Implementation: OMISSION: This weakness is caused by missing a security tactic during the architecture and design phase.

**•** Operation: OMISSION: This weakness is caused by missing a security tactic during the architecture and design phase.

## Common Consequences

**•** Impact: Read Files or Directories, Modify Files or Directories — Notes:

## Potential Mitigations

**•** Implementation: When storing data in the cloud (e.g., S3 buckets, Azure blobs, Google Cloud Storage, etc.), use the provider's controls to disable public access. (Effectiveness: N/A)

## Applicable Platforms

**•** None (Class: Not Language-Specific, Prevalence: Undetermined)

## Demonstrative Examples

**•** However, "Allow Blob Public Access" is set to true, meaning that anonymous/public users can access blobs.

**•** Suppose the command returns the following result: