# CWE Detail – CWE-565

## Description

The product relies on the existence or values of cookies when performing security-critical operations, but it does not properly ensure that the setting is valid for the associated user.

## Extended Description

N/A

## Threat-Mapped Scoring

Score: 0.0

Priority: Unclassified

## Observed Examples (CVEs)

**•** CVE-2008-5784: e-dating application allows admin privileges by setting the admin cookie to 1.

## Related Attack Patterns (CAPEC)

* CAPEC-226
* CAPEC-31
* CAPEC-39

## Attack TTPs

**•** T1539: Steal Web Session Cookie (Tactics: credential-access)

## Modes of Introduction

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

**•** Implementation: N/A

## Common Consequences

**•** Impact: Modify Application Data, Execute Unauthorized Code or Commands — Notes: Attackers can easily modify cookies, within the browser or by implementing the client-side code outside of the browser. Reliance on cookies without detailed validation and integrity checking can allow attackers to bypass authentication, conduct injection attacks such as SQL injection and cross-site scripting, or otherwise modify inputs in unexpected ways.

**•** Impact: Gain Privileges or Assume Identity — Notes: It is dangerous to use cookies to set a user's privileges. The cookie can be manipulated to escalate an attacker's privileges to an administrative level.

## Potential Mitigations

**•** Architecture and Design: Avoid using cookie data for a security-related decision. (Effectiveness: N/A)

**•** Implementation: Perform thorough input validation (i.e.: server side validation) on the cookie data if you're going to use it for a security related decision. (Effectiveness: N/A)

**•** Architecture and Design: Add integrity checks to detect tampering. (Effectiveness: N/A)

**•** Architecture and Design: Protect critical cookies from replay attacks, since cross-site scripting or other attacks may allow attackers to steal a strongly-encrypted cookie that also passes integrity checks. This mitigation applies to cookies that should only be valid during a single transaction or session. By enforcing timeouts, you may limit the scope of an attack. As part of your integrity check, use an unpredictable, server-side value that is not exposed to the client. (Effectiveness: N/A)

## Demonstrative Examples

**•** It is easy for an attacker to modify the "role" value found in the locally stored cookie, allowing privilege escalation.

## Notes

**•** Relationship: This problem can be primary to many types of weaknesses in web applications. A developer may perform proper validation against URL parameters while assuming that attackers cannot modify cookies. As a result, the program might skip basic input validation to enable cross-site scripting, SQL injection, price tampering, and other attacks..