# CWE Detail – CWE-640

## Description

The product contains a mechanism for users to recover or change their passwords without knowing the original password, but the mechanism is weak.

## Extended Description

It is common for an application to have a mechanism that provides a means for a user to gain access to their account in the event they forget their password. Very often the password recovery mechanism is weak, which has the effect of making it more likely that it would be possible for a person other than the legitimate system user to gain access to that user's account. Weak password recovery schemes completely undermine a strong password authentication scheme. This weakness may be that the security question is too easy to guess or find an answer to (e.g. because the question is too common, or the answers can be found using social media). Or there might be an implementation weakness in the password recovery mechanism code that may for instance trick the system into e-mailing the new password to an e-mail account other than that of the user. There might be no throttling done on the rate of password resets so that a legitimate user can be denied service by an attacker if an attacker tries to recover their password in a rapid succession. The system may send the original password to the user rather than generating a new temporary password. In summary, password recovery functionality, if not carefully designed and implemented can often become the system's weakest link that can be misused in a way that would allow an attacker to gain unauthorized access to the system.

## Threat-Mapped Scoring

Score: 3.0

Priority: P2 - Serious (High)

## Related Attack Patterns (CAPEC)

* CAPEC-50

## Modes of Introduction

**•** Architecture and Design: COMMISSION: This weakness refers to an incorrect design related to an architectural security tactic.

**•** Implementation: N/A

## Common Consequences

**•** Impact: Gain Privileges or Assume Identity — Notes: An attacker could gain unauthorized access to the system by retrieving legitimate user's authentication credentials.

**•** Impact: DoS: Resource Consumption (Other) — Notes: An attacker could deny service to legitimate system users by launching a brute force attack on the password recovery mechanism using user ids of legitimate users.

**•** Impact: Other — Notes: The system's security functionality is turned against the system by the attacker.

## Potential Mitigations

**•** Architecture and Design: Make sure that all input supplied by the user to the password recovery mechanism is thoroughly filtered and validated. (Effectiveness: N/A)

**•** Architecture and Design: Do not use standard weak security questions and use several security questions. (Effectiveness: N/A)

**•** Architecture and Design: Make sure that there is throttling on the number of incorrect answers to a security question. Disable the password recovery functionality after a certain (small) number of incorrect guesses. (Effectiveness: N/A)

**•** Architecture and Design: Require that the user properly answers the security question prior to resetting their password and sending the new password to the e-mail address of record. (Effectiveness: N/A)

**•** Architecture and Design: Never allow the user to control what e-mail address the new password will be sent to in the password recovery mechanism. (Effectiveness: N/A)

**•** Architecture and Design: Assign a new temporary password rather than revealing the original password. (Effectiveness: N/A)

## Applicable Platforms

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

## Demonstrative Examples

**•** N/A

## Notes

**•** Maintenance: This entry might be reclassified as a category or "loose composite," since it lists multiple specific errors that can make the mechanism weak. However, under view 1000, it could be a weakness under protection mechanism failure, although it is different from most PMF issues since it is related to a feature that is designed to bypass a protection mechanism (specifically, the lack of knowledge of a password).

**•** Maintenance: This entry probably needs to be split; see extended description.