# CWE Detail – CWE-307

## Description

The product does not implement sufficient measures to prevent multiple failed authentication attempts within a short time frame.

## Extended Description

N/A

## Threat-Mapped Scoring

Score: 0.0

Priority: Unclassified

## Observed Examples (CVEs)

**•** CVE-2019-0039: the REST API for a network OS has a high limit for number of connections, allowing brute force password guessing

**•** CVE-1999-1152: Product does not disconnect or timeout after multiple failed logins.

**•** CVE-2001-1291: Product does not disconnect or timeout after multiple failed logins.

**•** CVE-2001-0395: Product does not disconnect or timeout after multiple failed logins.

**•** CVE-2001-1339: Product does not disconnect or timeout after multiple failed logins.

**•** CVE-2002-0628: Product does not disconnect or timeout after multiple failed logins.

**•** CVE-1999-1324: User accounts not disabled when they exceed a threshold; possibly a resultant problem.

## Related Attack Patterns (CAPEC)

* CAPEC-16
* CAPEC-49
* CAPEC-560
* CAPEC-565
* CAPEC-600
* CAPEC-652
* CAPEC-653

## Attack TTPs

**•** T1110.001: Password Guessing (Tactics: credential-access)

**•** T1558: Steal or Forge Kerberos Tickets (Tactics: credential-access)

**•** T1110.003: Password Spraying (Tactics: credential-access)

**•** T1078: Valid Accounts (Tactics: defense-evasion, persistence, privilege-escalation, initial-access)

**•** T1110.004: Credential Stuffing (Tactics: credential-access)

## Modes of Introduction

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

## Common Consequences

**•** Impact: Bypass Protection Mechanism — Notes: An attacker could perform an arbitrary number of authentication attempts using different passwords, and eventually gain access to the targeted account using a brute force attack.

## Potential Mitigations

**•** Architecture and Design: Common protection mechanisms include: Disconnecting the user after a small number of failed attempts Implementing a timeout Locking out a targeted account Requiring a computational task on the user's part. (Effectiveness: N/A)

**•** Architecture and Design: Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid. Consider using libraries with authentication capabilities such as OpenSSL or the ESAPI Authenticator. [REF-45] (Effectiveness: N/A)

## Applicable Platforms

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

## Demonstrative Examples

**•** N/A

**•** However, the software makes no attempt to restrict excessive authentication attempts.

**•** However, there is no limit on parallel connections, so this does not increase the amount of time an attacker needs to complete an attack.

**•** The validateUser method will continuously check for a valid username and password without any restriction on the number of authentication attempts made. The method should limit the number of authentication attempts made to prevent brute force attacks as in the following example code.

**•** N/A