# CWE Detail – CWE-1253

## Description

The logic level used to set a system to a secure state relies on a fuse being unblown. An attacker can set the system to an insecure state merely by blowing the fuse.

## Extended Description

Fuses are often used to store secret data, including security configuration data. When not blown, a fuse is considered to store a logic 0, and, when blown, it indicates a logic 1. Fuses are generally considered to be one-directional, i.e., once blown to logic 1, it cannot be reset to logic 0. However, if the logic used to determine system-security state (by leveraging the values sensed from the fuses) uses negative logic, an attacker might blow the fuse and drive the system to an insecure state.

## Threat-Mapped Scoring

Score: 0.0

Priority: Unclassified

## Related Attack Patterns (CAPEC)

* CAPEC-74

## Modes of Introduction

**•** Architecture and Design: N/A

**•** Implementation: N/A

## Common Consequences

**•** Impact: Bypass Protection Mechanism, Gain Privileges or Assume Identity — Notes:

**•** Impact: DoS: Crash, Exit, or Restart — Notes:

**•** Impact: Read Memory — Notes:

**•** Impact: Modify Memory, Execute Unauthorized Code or Commands — Notes:

## Potential Mitigations

**•** Architecture and Design: Logic should be designed in a way that blown fuses do not put the product into an insecure state that can be leveraged by an attacker. (Effectiveness: N/A)

## Applicable Platforms

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

## Demonstrative Examples

**•** N/A

## Notes

**•** Maintenance: This entry is still under development and will continue to see updates and content improvements.