# CWE Detail – CWE-1223

## Description

A write-once register in hardware design is programmable by an untrusted software component earlier than the trusted software component, resulting in a race condition issue.

## Extended Description

Integrated circuits and hardware IP software programmable controls and settings are commonly stored in register circuits. These register contents have to be initialized at hardware reset to defined default values that are hard coded in the hardware description language (HDL) code of the hardware unit. A common security protection method used to protect register settings from modification by software is to make them write-once. This means the hardware implementation only allows writing to such registers once, and they become read-only after having been written once by software. This is useful to allow initial boot software to configure systems settings to secure values while blocking runtime software from modifying such hardware settings. Implementation issues in hardware design of such controls can expose such registers to a race condition security flaw. For example, consider a hardware design that has two different software/firmware modules executing in parallel. One module is trusted (module A) and another is untrusted (module B). In this design it could be possible for Module B to send write cycles to the write-once register before Module A. Since the field is write-once the programmed value from Module A will be ignored and the pre-empted value programmed by Module B will be used by hardware.

## Threat-Mapped Scoring

Score: 0.0

Priority: Unclassified

## Related Attack Patterns (CAPEC)

* CAPEC-26

## Modes of Introduction

**•** Architecture and Design: This weakness can appear in designs that use register write-once attributes with two or more software/firmware modules with varying levels of trust executing in parallel.

## Common Consequences

**•** Impact: Bypass Protection Mechanism — Notes: System configuration cannot be programmed in a secure way.

## Potential Mitigations

**•** Architecture and Design: During hardware design all register write-once or sticky fields must be evaluated for proper configuration. (Effectiveness: N/A)

**•** Testing: The testing phase should use automated tools to test that values are not reprogrammable and that write-once fields lock on writing zeros. (Effectiveness: N/A)

## Applicable Platforms

**•** Verilog (Class: None, Prevalence: Undetermined)

**•** VHDL (Class: None, Prevalence: Undetermined)

## Demonstrative Examples

**•** The first system component that sends a write cycle to this register can program the value. This could result in a race condition security issue in the SoC design, if an untrusted agent is running in the system in parallel with the trusted component that is expected to program the register.