# CWE Detail – CWE-1419

## Description

The product attempts to initialize a resource but does not correctly do so, which might leave the resource in an unexpected, incorrect, or insecure state when it is accessed.

## Extended Description

This can have security implications when the associated resource is expected to have certain properties or values. Examples include a variable that determines whether a user has been authenticated or not, or a register or fuse value that determines the security state of the product. For software, this weakness can frequently occur when implicit initialization is used, meaning the resource is not explicitly set to a specific value. For example, in C, memory is not necessarily cleared when it is allocated on the stack, and many scripting languages use a default empty, null value, or zero value when a variable is not explicitly initialized. For hardware, this weakness frequently appears with reset values and fuses. After a product reset, hardware may initialize registers incorrectly. During different phases of a product lifecycle, fuses may be set to incorrect values. Even if fuses are set to correct values, the lines to the fuse could be broken or there might be hardware on the fuse line that alters the fuse value to be incorrect.

## Threat-Mapped Scoring

Score: 1.8

Priority: P4 - Informational (Low)

## Observed Examples (CVEs)

**•** CVE-2020-27211: Chain: microcontroller system-on-chip uses a register value stored in flash to set product protection state on the memory bus and does not contain protection against fault injection (CWE-1319) which leads to an incorrect initialization of the memory bus (CWE-1419) causing the product to be in an unprotected state.

**•** CVE-2023-25815: chain: a change in an underlying package causes the gettext function to use implicit initialization with a hard-coded path (CWE-1419) under the user-writable C:\ drive, introducing an untrusted search path element (CWE-427) that enables spoofing of messages.

**•** CVE-2022-43468: WordPress module sets internal variables based on external inputs, allowing false reporting of the number of views

**•** CVE-2022-36349: insecure default variable initialization in BIOS firmware for a hardware board allows DoS

**•** CVE-2015-7763: distributed filesystem only initializes part of the variable-length padding for a packet, allowing attackers to read sensitive information from previously-sent packets in the same memory location

## Modes of Introduction

**•** Implementation: N/A

**•** Manufacturing: N/A

**•** Installation: N/A

**•** System Configuration: N/A

**•** Operation: N/A

## Common Consequences

**•** Impact: Read Memory, Read Application Data, Unexpected State — Notes:

**•** Impact: Gain Privileges or Assume Identity — Notes:

**•** Impact: Varies by Context — Notes: The technical impact can vary widely based on how the resource is used in the product, and whether its contents affect security decisions.

## Potential Mitigations

**•** Implementation: Choose the safest-possible initialization for security-related resources. (Effectiveness: N/A)

**•** Implementation: Ensure that each resource (whether variable, memory buffer, register, etc.) is fully initialized. (Effectiveness: N/A)

**•** Implementation: Pay close attention to complex conditionals or reset sources that affect initialization, since some paths might not perform the initialization. (Effectiveness: N/A)

**•** Architecture and Design: Ensure that the design and architecture clearly identify what the initialization should be, and that the initialization does not have security implications. (Effectiveness: N/A)

## Applicable Platforms

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

## Demonstrative Examples

**•** These example instantiations show how, in a hardware design, it would be possible to instantiate the register module with insecure defaults and parameters.

**•** Because the $authorized variable is never initialized, PHP will automatically set $authorized to any value included in the POST request if register\_globals is enabled. An attacker can send a POST request with an unexpected third value 'authorized' set to 'true' and gain authorized status without supplying valid credentials.

**•** However, in the buggy SoC, these registers are all enabled at reset, i.e., essentially granting unrestricted access to all system resources [REF-1438]. This will introduce security vulnerabilities and risks to the system, such as privilege escalation or exposing sensitive information to unauthorized users or processes.