# CWE Detail – CWE-454

## Description

The product initializes critical internal variables or data stores using inputs that can be modified by untrusted actors.

## Extended Description

A product system should be reluctant to trust variables that have been initialized outside of its trust boundary, especially if they are initialized by users. The variables may have been initialized incorrectly. If an attacker can initialize the variable, then they can influence what the vulnerable system will do.

## Threat-Mapped Scoring

Score: 0.0

Priority: Unclassified

## Observed Examples (CVEs)

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

**•** CVE-2000-0959: Does not clear dangerous environment variables, enabling symlink attack.

**•** CVE-2001-0033: Specify alternate configuration directory in environment variable, enabling untrusted path.

**•** CVE-2001-0872: Dangerous environment variable not cleansed.

**•** CVE-2001-0084: Specify arbitrary modules using environment variable.

## Modes of Introduction

**•** Architecture and Design: N/A

**•** Implementation: N/A

## Common Consequences

**•** Impact: Modify Application Data — Notes: An attacker could gain access to and modify sensitive data or system information.

## Potential Mitigations

**•** Implementation: A product system should be reluctant to trust variables that have been initialized outside of its trust boundary. Ensure adequate checking (e.g. input validation) is performed when relying on input from outside a trust boundary. (Effectiveness: N/A)

**•** Architecture and Design: Avoid any external control of variables. If necessary, restrict the variables that can be modified using an allowlist, and use a different namespace or naming convention if possible. (Effectiveness: N/A)

## Applicable Platforms

**•** PHP (Class: None, Prevalence: Sometimes)

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

## Demonstrative Examples

**•** If an attacker is able to modify the system property, then it may be possible to coax the application into divulging sensitive information by virtue of the fact that additional debug information is printed/exposed as the debug level increases.

**•** Any user can activate the debug mode, gaining administrator privileges. An attacker may also use the information printed by the phpinfo() function to further exploit the system. .

## Notes

**•** Relationship: Overlaps Missing variable initialization, especially in PHP.

**•** Applicable Platform: This is often found in PHP due to register\_globals and the common practice of storing library/include files under the web document root so that they are available using a direct request.