# CWE Detail – CWE-909

## Description

The product does not initialize a critical resource.

## Extended Description

Many resources require initialization before they can be properly used. If a resource is not initialized, it could contain unpredictable or expired data, or it could be initialized to defaults that are invalid. This can have security implications when the resource is expected to have certain properties or values.

## Threat-Mapped Scoring

Score: 1.8

Priority: P4 - Informational (Low)

## Observed Examples (CVEs)

**•** CVE-2020-20739: A variable that has its value set in a conditional statement is sometimes used when the conditional fails, sometimes causing data leakage

**•** CVE-2005-1036: Chain: Bypass of access restrictions due to improper authorization (CWE-862) of a user results from an improperly initialized (CWE-909) I/O permission bitmap

## Modes of Introduction

**•** Implementation: N/A

## Common Consequences

**•** Impact: Read Memory, Read Application Data — Notes: When reusing a resource such as memory or a program variable, the original contents of that resource may not be cleared before it is sent to an untrusted party.

**•** Impact: DoS: Crash, Exit, or Restart — Notes: The uninitialized resource may contain values that cause program flow to change in ways that the programmer did not intend.

## Potential Mitigations

**•** Implementation: Explicitly initialize the resource before use. If this is performed through an API function or standard procedure, follow all specified steps. (Effectiveness: N/A)

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

**•** Implementation: Avoid race conditions (CWE-362) during initialization routines. (Effectiveness: N/A)

**•** Build and Compilation: Run or compile your product with settings that generate warnings about uninitialized variables or data. (Effectiveness: N/A)

## Applicable Platforms

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

## Demonstrative Examples

**•** N/A

**•** If the application is unable to extract the state information - say, due to a database timeout - then the $uid variable will not be explicitly set by the programmer. This will cause $uid to be regarded as equivalent to "0" in the conditional, allowing the original user to perform administrator actions. Even if the attacker cannot directly influence the state data, unexpected errors could cause incorrect privileges to be assigned to a user just by accident.

**•** This might seem innocent enough, but str was not initialized, so it contains random memory. As a result, str[0] might not contain the null terminator, so the copy might start at an offset other than 0. The consequences can vary, depending on the underlying memory.

**•** When the printf() is reached,  
 test\_string might be an unexpected address, so the  
 printf might print junk strings (CWE-457). To fix this code, there are a couple approaches to  
 making sure that test\_string has been properly set once  
 it reaches the printf(). One solution would be to set test\_string to an  
 acceptable default before the conditional: