# CWE Detail – CWE-824

## Description

The product accesses or uses a pointer that has not been initialized.

## Extended Description

If the pointer contains an uninitialized value, then the value might not point to a valid memory location. This could cause the product to read from or write to unexpected memory locations, leading to a denial of service. If the uninitialized pointer is used as a function call, then arbitrary functions could be invoked. If an attacker can influence the portion of uninitialized memory that is contained in the pointer, this weakness could be leveraged to execute code or perform other attacks. Depending on memory layout, associated memory management behaviors, and product operation, the attacker might be able to influence the contents of the uninitialized pointer, thus gaining more fine-grained control of the memory location to be accessed.

## Threat-Mapped Scoring

Score: 1.5

Priority: P4 - Informational (Low)

## Observed Examples (CVEs)

**•** CVE-2024-32878: LLM product has a free of an uninitialized pointer

**•** CVE-2019-3836: Chain: secure communications library does not initialize a local variable for a data structure (CWE-456), leading to access of an uninitialized pointer (CWE-824).

**•** CVE-2018-14641: Chain: C union member is not initialized (CWE-456), leading to access of invalid pointer (CWE-824)

**•** CVE-2010-0211: chain: unchecked return value (CWE-252) leads to free of invalid, uninitialized pointer (CWE-824).

**•** CVE-2009-2768: Pointer in structure is not initialized, leading to NULL pointer dereference (CWE-476) and system crash.

**•** CVE-2009-1721: Free of an uninitialized pointer.

**•** CVE-2009-1415: Improper handling of invalid signatures leads to free of invalid pointer.

**•** CVE-2009-0846: Invalid encoding triggers free of uninitialized pointer.

**•** CVE-2009-0040: Crafted PNG image leads to free of uninitialized pointer.

**•** CVE-2008-2934: Crafted GIF image leads to free of uninitialized pointer.

**•** CVE-2007-4682: Access of uninitialized pointer might lead to code execution.

**•** CVE-2007-4639: Step-based manipulation: invocation of debugging function before the primary initialization function leads to access of an uninitialized pointer and code execution.

**•** CVE-2007-4000: Unchecked return values can lead to a write to an uninitialized pointer.

**•** CVE-2007-2442: zero-length input leads to free of uninitialized pointer.

**•** CVE-2007-1213: Crafted font leads to uninitialized function pointer.

**•** CVE-2006-6143: Uninitialized function pointer in freed memory is invoked

**•** CVE-2006-4175: LDAP server mishandles malformed BER queries, leading to free of uninitialized memory

**•** CVE-2006-0054: Firewall can crash with certain ICMP packets that trigger access of an uninitialized pointer.

**•** CVE-2003-1201: LDAP server does not initialize members of structs, which leads to free of uninitialized pointer if an LDAP request fails.

## Common Consequences

**•** Impact: Read Memory — Notes: If the uninitialized pointer is used in a read operation, an attacker might be able to read sensitive portions of memory.

**•** Impact: DoS: Crash, Exit, or Restart — Notes: If the uninitialized pointer references a memory location that is not accessible to the product, or points to a location that is "malformed" (such as NULL) or larger than expected by a read or write operation, then a crash may occur.

**•** Impact: Execute Unauthorized Code or Commands — Notes: If the uninitialized pointer is used in a function call, or points to unexpected data in a write operation, then code execution may be possible.

## Applicable Platforms

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

**•** C++ (Class: None, Prevalence: Undetermined)

## Notes

**•** Maintenance: There are close relationships between incorrect pointer dereferences and other weaknesses related to buffer operations. There may not be sufficient community agreement regarding these relationships. Further study is needed to determine when these relationships are chains, composites, perspective/layering, or other types of relationships. As of September 2010, most of the relationships are being captured as chains.

**•** Terminology: Many weaknesses related to pointer dereferences fall under the general term of "memory corruption" or "memory safety." As of September 2010, there is no commonly-used terminology that covers the lower-level variants.