# CWE Detail – CWE-696

## Description

The product performs multiple related behaviors, but the behaviors are performed in the wrong order in ways which may produce resultant weaknesses.

## Extended Description

N/A

## Threat-Mapped Scoring

Score: 0.0

Priority: Unclassified

## Observed Examples (CVEs)

**•** CVE-2019-9805: Chain: Creation of the packet client occurs before initialization is complete (CWE-696) resulting in a read from uninitialized memory (CWE-908), causing memory corruption.

**•** CVE-2007-5191: file-system management programs call the setuid and setgid functions in the wrong order and do not check the return values, allowing attackers to gain unintended privileges

**•** CVE-2007-1588: C++ web server program calls Process::setuid before calling Process::setgid, preventing it from dropping privileges, potentially allowing CGI programs to be called with higher privileges than intended

**•** CVE-2022-37734: Chain: lexer in Java-based GraphQL server does not enforce maximum of tokens early enough (CWE-696), allowing excessive CPU consumption (CWE-1176)

## Related Attack Patterns (CAPEC)

* CAPEC-463

## Modes of Introduction

**•** Architecture and Design: N/A

**•** Implementation: N/A

## Common Consequences

**•** Impact: Alter Execution Logic — Notes:

## Demonstrative Examples

**•** The problem with the above code is that the validation step occurs before canonicalization occurs. An attacker could provide an input path of "/safe\_dir/../" that would pass the validation step. However, the canonicalization process sees the double dot as a traversal to the parent directory and hence when canonicized the path would become just "/".

**•** This code first reads a specified file into memory, then prints the file if the user is authorized to see its contents. The read of the file into memory may be resource intensive and is unnecessary if the user is not allowed to see the file anyway.

**•** This code uses Verilog blocking assignments for data\_out and grant\_access. Therefore, these assignments happen sequentially (i.e., data\_out is updated to new value first, and grant\_access is updated the next cycle) and not in parallel. Therefore, the asset data\_out is allowed to be modified even before the access control check is complete and grant\_access signal is set. Since grant\_access does not have a reset value, it will be meta-stable and will randomly go to either 0 or 1.