# CWE Detail – CWE-181

## Description

The product validates data before it has been filtered, which prevents the product from detecting data that becomes invalid after the filtering step.

## Extended Description

This can be used by an attacker to bypass the validation and launch attacks that expose weaknesses that would otherwise be prevented, such as injection.

## Threat-Mapped Scoring

Score: 0.0

Priority: Unclassified

## Observed Examples (CVEs)

**•** CVE-2002-0934: Directory traversal vulnerability allows remote attackers to read or modify arbitrary files via invalid characters between two . (dot) characters, which are filtered and result in a ".." sequence.

**•** CVE-2003-0282: Directory traversal vulnerability allows attackers to overwrite arbitrary files via invalid characters between two . (dot) characters, which are filtered and result in a ".." sequence.

## Related Attack Patterns (CAPEC)

* CAPEC-120
* CAPEC-267
* CAPEC-3
* CAPEC-43
* CAPEC-78
* CAPEC-79
* CAPEC-80

## Attack TTPs

**•** T1027: Obfuscated Files or Information (Tactics: defense-evasion)

## Modes of Introduction

**•** Implementation: N/A

## Common Consequences

**•** Impact: Bypass Protection Mechanism — Notes:

## Potential Mitigations

**•** Implementation: Inputs should be decoded and canonicalized to the application's current internal representation before being filtered. (Effectiveness: N/A)

## Applicable Platforms

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

## Demonstrative Examples

**•** While the script attempts to screen for '..' sequences, an attacker can submit a directory path including ".~.", which will then become ".." after the filtering step. This allows a Path Traversal (CWE-21) attack to occur.

## Notes

**•** Research Gap: This category is probably under-studied.