# CWE Detail – CWE-641

## Description

The product constructs the name of a file or other resource using input from an upstream component, but it does not restrict or incorrectly restricts the resulting name.

## Extended Description

This may produce resultant weaknesses. For instance, if the names of these resources contain scripting characters, it is possible that a script may get executed in the client's browser if the application ever displays the name of the resource on a dynamically generated web page. Alternately, if the resources are consumed by some application parser, a specially crafted name can exploit some vulnerability internal to the parser, potentially resulting in execution of arbitrary code on the server machine. The problems will vary based on the context of usage of such malformed resource names and whether vulnerabilities are present in or assumptions are made by the targeted technology that would make code execution possible.

## Threat-Mapped Scoring

Score: 1.8

Priority: P4 - Informational (Low)

## Modes of Introduction

**•** Architecture and Design: N/A

**•** Implementation: REALIZATION: This weakness is caused during implementation of an architectural security tactic.

## Common Consequences

**•** Impact: Execute Unauthorized Code or Commands — Notes: Execution of arbitrary code in the context of usage of the resources with dangerous names.

**•** Impact: Read Application Data, DoS: Crash, Exit, or Restart — Notes: Crash of the consumer code of these resources resulting in information leakage or denial of service.

## Potential Mitigations

**•** Architecture and Design: Do not allow users to control names of resources used on the server side. (Effectiveness: N/A)

**•** Architecture and Design: Perform allowlist input validation at entry points and also before consuming the resources. Reject bad file names rather than trying to cleanse them. (Effectiveness: N/A)

**•** Architecture and Design: Make sure that technologies consuming the resources are not vulnerable (e.g. buffer overflow, format string, etc.) in a way that would allow code execution if the name of the resource is malformed. (Effectiveness: N/A)

## Applicable Platforms

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