# CWE Detail – CWE-647

## Description

The product defines policy namespaces and makes authorization decisions based on the assumption that a URL is canonical. This can allow a non-canonical URL to bypass the authorization.

## Extended Description

If an application defines policy namespaces and makes authorization decisions based on the URL, but it does not require or convert to a canonical URL before making the authorization decision, then it opens the application to attack. For example, if the application only wants to allow access to http://www.example.com/mypage, then the attacker might be able to bypass this restriction using equivalent URLs such as: http://WWW.EXAMPLE.COM/mypage http://www.example.com/%6Dypage (alternate encoding) http://192.168.1.1/mypage (IP address) http://www.example.com/mypage/ (trailing /) http://www.example.com:80/mypage Therefore it is important to specify access control policy that is based on the path information in some canonical form with all alternate encodings rejected (which can be accomplished by a default deny rule).

## Threat-Mapped Scoring

Score: 0.0

Priority: Unclassified

## Modes of Introduction

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

**•** Operation: N/A

## Common Consequences

**•** Impact: Bypass Protection Mechanism — Notes: An attacker may be able to bypass the authorization mechanism to gain access to the otherwise-protected URL.

**•** Impact: Read Files or Directories — Notes: If a non-canonical URL is used, the server may choose to return the contents of the file, instead of pre-processing the file (e.g. as a program).

## Potential Mitigations

**•** Architecture and Design: Make access control policy based on path information in canonical form. Use very restrictive regular expressions to validate that the path is in the expected form. (Effectiveness: N/A)

**•** Architecture and Design: Reject all alternate path encodings that are not in the expected canonical form. (Effectiveness: N/A)

## Applicable Platforms

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

## Demonstrative Examples

**•** N/A