# CWE Detail – CWE-425

## Description

The web application does not adequately enforce appropriate authorization on all restricted URLs, scripts, or files.

## Extended Description

Web applications susceptible to direct request attacks often make the false assumption that such resources can only be reached through a given navigation path and so only apply authorization at certain points in the path.

## Threat-Mapped Scoring

Score: 1.8

Priority: P4 - Informational (Low)

## Observed Examples (CVEs)

**•** CVE-2022-29238: Access-control setting in web-based document collaboration tool is not properly implemented by the code, which prevents listing hidden directories but does not prevent direct requests to files in those directories.

**•** CVE-2022-23607: Python-based HTTP library did not scope cookies to a particular domain such that "supercookies" could be sent to any domain on redirect.

**•** CVE-2004-2144: Bypass authentication via direct request.

**•** CVE-2005-1892: Infinite loop or infoleak triggered by direct requests.

**•** CVE-2004-2257: Bypass auth/auth via direct request.

**•** CVE-2005-1688: Direct request leads to infoleak by error.

**•** CVE-2005-1697: Direct request leads to infoleak by error.

**•** CVE-2005-1698: Direct request leads to infoleak by error.

**•** CVE-2005-1685: Authentication bypass via direct request.

**•** CVE-2005-1827: Authentication bypass via direct request.

**•** CVE-2005-1654: Authorization bypass using direct request.

**•** CVE-2005-1668: Access privileged functionality using direct request.

**•** CVE-2002-1798: Upload arbitrary files via direct request.

## Related Attack Patterns (CAPEC)

* CAPEC-127
* CAPEC-143
* CAPEC-144
* CAPEC-668
* CAPEC-87

## Attack TTPs

**•** T1083: File and Directory Discovery (Tactics: discovery)

**•** T1565.002: Transmitted Data Manipulation (Tactics: impact)

## Modes of Introduction

**•** Implementation: N/A

**•** Operation: N/A

## Common Consequences

**•** Impact: Read Application Data, Modify Application Data, Execute Unauthorized Code or Commands, Gain Privileges or Assume Identity — Notes:

## Potential Mitigations

**•** Architecture and Design: Apply appropriate access control authorizations for each access to all restricted URLs, scripts or files. (Effectiveness: N/A)

**•** Architecture and Design: Consider using MVC based frameworks such as Struts. (Effectiveness: N/A)

## Applicable Platforms

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

## Demonstrative Examples

**•** N/A

## Notes

**•** Relationship: Overlaps Modification of Assumed-Immutable Data (MAID), authorization errors, container errors; often primary to other weaknesses such as XSS and SQL injection.

**•** Theoretical: "Forced browsing" is a step-based manipulation involving the omission of one or more steps, whose order is assumed to be immutable. The application does not verify that the first step was performed successfully before the second step. The consequence is typically "authentication bypass" or "path disclosure," although it can be primary to all kinds of weaknesses, especially in languages such as PHP, which allow external modification of assumed-immutable variables.