# CWE Detail – CWE-610

## Description

The product uses an externally controlled name or reference that resolves to a resource that is outside of the intended control sphere.

## Extended Description

N/A

## Threat-Mapped Scoring

Score: 1.8

Priority: P4 - Informational (Low)

## Observed Examples (CVEs)

**•** CVE-2022-3032: An email client does not block loading of remote objects in a nested document.

**•** CVE-2022-45918: Chain: a learning management tool debugger uses external input to locate previous session logs (CWE-73) and does not properly validate the given path (CWE-20), allowing for filesystem path traversal using "../" sequences (CWE-24)

**•** CVE-2018-1000613: Cryptography API uses unsafe reflection when deserializing a private key

**•** CVE-2020-11053: Chain: Go-based Oauth2 reverse proxy can send the authenticated user to another site at the end of the authentication flow. A redirect URL with HTML-encoded whitespace characters can bypass the validation (CWE-1289) to redirect to a malicious site (CWE-601)

**•** CVE-2022-42745: Recruiter software allows reading arbitrary files using XXE

**•** CVE-2004-2331: Database system allows attackers to bypass sandbox restrictions by using the Reflection API.

## Related Attack Patterns (CAPEC)

* CAPEC-219

## Modes of Introduction

**•** Architecture and Design: COMMISSION: This weakness refers to an incorrect design related to an architectural security tactic.

## Common Consequences

**•** Impact: Read Application Data, Modify Application Data — Notes: An adversary could read or modify data, depending on how the resource is intended to be used.

**•** Impact: Gain Privileges or Assume Identity — Notes: An adversary that can supply a reference to an unintended resource can potentially access a resource that they do not have privileges for, thus bypassing existing access control mechanisms.

## Demonstrative Examples

**•** The problem with this Java servlet code is that an attacker could use the RedirectServlet as part of an e-mail phishing scam to redirect users to a malicious site. An attacker could send an HTML formatted e-mail directing the user to log into their account by including in the e-mail the following link:

## Notes

**•** Relationship: This is a general class of weakness, but most research is focused on more specialized cases, such as path traversal (CWE-22) and symlink following (CWE-61). A symbolic link has a name; in general, it appears like any other file in the file system. However, the link includes a reference to another file, often in another directory - perhaps in another sphere of control. Many common library functions that accept filenames will "follow" a symbolic link and use the link's target instead.

**•** Maintenance: The relationship between CWE-99 and CWE-610 needs further investigation and clarification. They might be duplicates. CWE-99 "Resource Injection," as originally defined in Seven Pernicious Kingdoms taxonomy, emphasizes the "identifier used to access a system resource" such as a file name or port number, yet it explicitly states that the "resource injection" term does not apply to "path manipulation," which effectively identifies the path at which a resource can be found and could be considered to be one aspect of a resource identifier. Also, CWE-610 effectively covers any type of resource, whether that resource is at the system layer, the application layer, or the code layer.