What is path traversal?

Path traversal (directory traversal) is a vulnerability where an application builds a file path from user input without restricting it. An attacker then uses `../` sequences to escape the intended directory and read (or sometimes write) files elsewhere on the server, such as `/etc/passwd` or application secrets. The fix is to map choices to a server-side allow-list and confirm the resolved path stays inside an allowed base directory.

HOW IT WORKS

01 How it works and example

The attacker manipulates the filename or path parameter:

- Read a system file: `?file=../../../../etc/passwd`
- On Windows: `?file=..\..\..\windows\win.ini`
- Bypass naive filters with encoding (`%2e%2e%2f`), double encoding, or `....//` (which collapses to `../` after one round of stripping).
- Reach application config and secrets relative to the base directory.

Where the app also writes based on the path, traversal can overwrite files. Examples shown for defensive context.

> CANONICALISE, THEN CHECK
>
> *The reliable fix is to resolve the final absolute path (canonicalise) and verify it still starts with the allowed base directory. Blocking `../` strings alone is bypassable with encoding and tricks like `....//`.*

HOW TO DEFEND

- Avoid user-controlled paths. Map user choices to a server-side allow-list of files (an ID to a known filename), not a raw path.
- Canonicalise and confirm containment: resolve the absolute path and check it begins with the intended base directory before accessing it.
- Decode before validating so encoded traversal cannot slip through.
- Run with least privilege so the process cannot read sensitive files even if traversal occurs.
- Test every file parameter for traversal.

SOURCES

[1] OWASP: Path Traversal

[2] MITRE CWE-22: Improper Limitation of a Pathname to a Restricted Directory

[3] OWASP Top 10

Test your application for path traversal and 30+ other classes.

securelayer7.net/learn/application-security/path-traversal

Open online