# Path Traversal

**Path Traversal occurs when an application uses unsanitized user input to construct file or directory paths, allowing attackers to navigate outside the intended directory and:**

* **Access system files (e.g., /etc/passwd)**
* **Read source code or secrets**
* **Sometimes write files or achieve RCE (via LFI or log poisoning)**

# Example

<?php

$file = $\_GET['page'];

include("pages/" . $file);

?>

If attacker supplies: ?page=../../../../etc/passwd  
The server includes the system file outside pages/

# Impact

| Impact | Description |
| --- | --- |
| Unauthorized File Access | Read private files like .env, config.php, etc. |
| Credential Exposure | Extract passwords, DB creds from config files |
| LFI → RCE | Combine with log poisoning or shell upload |
| App Reconnaissance | Discover app structure, server layout |
| Security Misconfig Bypass | Access hidden or sensitive directories |

# Common Payloads

| Payload | Goal |
| --- | --- |
| ../../../../etc/passwd | Linux user list |
| ..\..\..\..\boot.ini | Windows system file |
| %2e%2e%2fetc/passwd | URL-encoded traversal |
| ..%c0%af../etc/passwd | Unicode-encoded |
| ....//....//....// | Bypass naive sanitization |
| ../../../uploads/shell.php | Execute file outside web root |

# Traversal Variants

| Type | Description |
| --- | --- |
| Relative Traversal | ../, ..\, etc. to go up the tree |
| Encoded Traversal | %2e%2e/, %2e%2e%5c, %252e%252e/ |
| Double Encoded | %252e%252e/ interpreted twice |
| Null Byte + Extension Bypass | ../../config.php%00.jpg (legacy PHP) |
| ZIP Path Traversal | Add ../ to file paths in archives → extract into parent dirs |

# Tools to Detect/Exploit

| Tool | Purpose |
| --- | --- |
| Burp Suite | Intercept file paths and test traversal |
| DotDotPwn | Brute-force traversal on parameters |
| wfuzz / ffuf | Path discovery and parameter fuzzing |
| Nikto / Nmap NSE | Detect known path disclosure issues |
| LFI Suite | If traversal leads to file inclusion |

# Testing Approach

**Identify Parameters**:  
Look for:

* file=, path=, template=, page=

**Try Basic Payloads**:  
?file=../../../../etc/passwd

**Try Encoded Payloads**:  
?file=%2e%2e%2f%2e%2e%2fetc%2fpasswd

**Look for File Indicators**:

* root:x: (Linux)
* [boot loader] (Windows)
* <php (code disclosure)

# Mitigation

## 1. Canonicalize and Validate Input

* Use realpath() or equivalent to **normalize paths**:

$realPath = realpath($userPath);

$basePath = realpath('/app/templates/');

if (strpos($realPath, $basePath) !== 0) {

die("Invalid file path!");

}

## 2. Whitelist File Names

Only allow specific filenames, not paths:

$allowed = ['home', 'about', 'contact'];

if (!in\_array($\_GET['page'], $allowed)) {

die("Invalid page");

}

## 3. Deny ../ and Encoded Versions

Block patterns:

../, ..\

%2e%2e, %252e, etc.

## 4. Set Least Privilege on File System

Web process should not have access to sensitive system files

Deny read access outside web root (/var/www/html)

## 5. Disable Directory Listing

Use .htaccess, nginx config to deny open access

Don’t expose backup files like .bak, .swp, etc.

# Points

“Path Traversal allows attackers to **escape the intended directory** using sequences like ../ to access unauthorized files.”

“Common targets include /etc/passwd, source code, .env, and uploaded files.”

“Mitigation involves **strict path sanitization**, **input validation**, and using safe filesystem APIs.”

# Real-World Exploits

| Company | Description |
| --- | --- |
| PayPal | Path traversal led to private file access (Bug Bounty) |
| Apache Tomcat (CVE-2020-1938) | Ghostcat exploit allowed traversal |
| Drupal | Path traversal allowed arbitrary file read in /sites/default/files |
| Android Zip Vulnerability | App ZIP extraction used ../ to overwrite app files |