# 🛡️ Local File Inclusion (LFI) - Rules Explained in Detail

## 🔎 What is LFI?

**Local File Inclusion (LFI)** is a web vulnerability where an attacker includes files on the local server by manipulating input parameters that refer to file paths.

### 📄 Example:

<?php include($\_GET['page']); ?>

Malicious request:

http://example.com/index.php?page=../../../../etc/passwd

## 🌪️ OWASP CRS - LFI Detection Rules in Detail

### ✅ Common Syntax Pattern

SecRule VARIABLES "@operator pattern" "id:X,phase:Y,deny,msg:'message',..."

### 🔜 Key Rule Components

* ARGS, REQUEST\_URI, etc.: Variables to inspect
* @rx, @pmf: Matching operators
* phase: ModSecurity phase
* msg: Log message
* severity: Risk level

## 🔹 Rule 930100 - Path Traversal Using ../

SecRule ARGS "@rx (\.\./)+" \  
 "id:930100,phase:2,deny,status:403,log,msg:'Path Traversal Attack (/../)',severity:CRITICAL"

### 💡 Purpose:

Detects repeated ../ sequences in request parameters.

### 🔎 Example:

?page=../../../etc/passwd

### ⚠️ False Positives:

?page=../images/logo.png

### 🔧 Fix:

SecRuleUpdateTargetById 930100 "!ARGS:page"

## 🔹 Rule 930110 - Encoded Traversal

SecRule ARGS "@rx (%2e%2e%2f|..%2f|%2e%2e/)" \  
 "id:930110,phase:2,deny,status:403,log,msg:'Encoded Traversal Attempt',severity:CRITICAL"

### 💡 Purpose:

Detects encoded ../ sequences.

### 🔎 Example:

?page=%2e%2e%2fetc%2fpasswd

### ⚠️ False Positives:

?page=..%2fmanual.pdf

### 🔧 Fix:

SecRuleUpdateTargetById 930110 "!ARGS:page"

## 🔹 Rule 930120 - OS File Access Attempt

SecRule ARGS|REQUEST\_URI "@pmf lfi-os-files.data" \  
 "id:930120,phase:2,deny,status:403,msg:'OS file access attempt',severity:CRITICAL"

### 💡 Purpose:

Blocks attempts to access OS files like /etc/passwd.

### 🔎 Example:

?page=/etc/passwd

### ⚠️ False Positives:

Rare (only if app serves legit OS-like filenames)

### 🔧 Fix:

SecRuleUpdateTargetById 930120 "!ARGS:page"

## 🔹 Rule 930130 - Restricted App File Access

SecRule REQUEST\_FILENAME "@pmf restricted-files.data" \  
 "id:930130,phase:2,deny,status:403,msg:'Restricted file access',severity:CRITICAL"

### 💡 Purpose:

Detects access to sensitive files like .env, .git/config, etc.

### 🔎 Example:

GET /.env

### ⚠️ False Positives:

Accessing safe files like .env.sample

### 🔧 Fix:

SecRuleUpdateTargetById 930130 "!REQUEST\_FILENAME:/safe/path/.env.sample"

## 🔮 Phase Overview

| Phase | What it does | Relevant LFI Rules |
| --- | --- | --- |
| 1 | Header parsing & PL checks | 930011, 930013, 930015, 930017 |
| 2 | Request body & parameters | 930100, 930110, 930120, 930130 |

## ⚖️ Where to Apply Updates

* modsecurity\_crs\_zz\_custom.conf: Best place to override rules
* Place overrides **after** CRS include line
* Use SecRuleUpdateTargetById or SecRuleRemoveById carefully

## 🚫 Avoid Full Rule Removal

SecRuleRemoveById 930100

Only use this if absolutely necessary. Better to exclude specific parameters.

## 🌐 Example: Safe File Serving in PHP

$allowed = ['manual.pdf', 'logo.png'];  
$file = $\_GET['file'];  
  
if (in\_array($file, $allowed)) {  
 readfile("files/" . $file);  
} else {  
 die("Access denied.");  
}

## ✅ Final Tips

* Keep PL1 LFI rules active
* Log false positives for review
* Whitelist only trusted parameters or paths
* Sanitize inputs in the app layer