What are file upload vulnerabilities?

File upload vulnerabilities arise when an application accepts uploaded files without adequately validating their type, content, and storage location. The highest-impact case is uploading a server-executable file, a web shell, into a directory the server will run, giving remote code execution. Weaker validation also enables stored cross-site scripting, path traversal, and denial of service. The root causes are trusting the client-supplied filename or content type and storing uploads where they can be executed.

HOW IT WORKS

01 The abuse and bypass tricks

Testers try to place an executable file past the app's checks. Shown for defensive context:

- Weak extension checks: try `shell.php`, then bypasses such as `shell.php.jpg`, `shell.pHp`, or alternate executable extensions like `.phtml-`, `.php5`, `.aspx`.
- Content-type spoofing: send a script but set the `Content-Type` header to `image/png`.
- Magic-byte prefixing: prepend real image header bytes so a content sniff passes, while the file still executes.
- Config file uploads: a crafted `.htaccess` can make the server treat new extensions as executable.
- Reach and run: once uploaded, browse to the file's URL to trigger execution.

Because a single successful upload can mean full compromise, upload handling is scrutinized closely during web application security testing.

HOW TO DEFEND

- Validate type by content, not by name: check the actual file with a trusted library, and allowlist expected types rather than blocklisting bad ones.
- Store uploads outside the web root or in object storage, and serve them through a handler that never executes them.
- Rename files to random identifiers and drop the user-supplied name and extension so path and execution tricks fail.
- Disable script execution in the upload directory at the server level as a backstop.
- Enforce size and rate limits, and scan content where appropriate.

SOURCES

[1] OWASP: File Upload Cheat Sheet

[2] PortSwigger: File upload vulnerabilities

[3] MITRE ATT&CK: Server Software Component: Web Shell (T1505.003)

Find the upload flaw before an attacker does.

securelayer7.net/learn/web-exploitation/what-are-file-upload-vulnerabilities

Open online