What is a web shell?

A web shell is a malicious script placed on a web server (PHP, ASPX, JSP, and others) that lets an attacker run commands on the server through a normal web request. It is a backdoor that survives reboots, runs with the web server's privileges, and reaches the server through ordinary HTTP/HTTPS, so it often passes through firewalls. Attackers plant web shells via file upload flaws, LFI, or other web vulnerabilities. It maps to MITRE T1505.003.

HOW IT WORKS

01 The technique and payload

The attacker gets a script into the web root and calls it:

- A minimal PHP shell: `<?php system($_GET["c"]); ?>` saved as `info.php`, then `https://site/info.php?c=id`.
- Planted via a file upload vulnerability, LFI log poisoning, or a compromised CMS plugin.
- Real-world web shells add authentication, file management, and obfuscation to evade detection.

From there the attacker runs commands, pivots, and escalates. Documented for defensive context.

> REACHABLE THROUGH THE FRONT DOOR
>
> *A web shell needs no new port or connection out, it answers normal web requests on the site's existing ports, so it often slips past firewalls. Finding it means inspecting the web files, not the network.*

HOW TO DEFEND

- Fix the entry points: validate file uploads, patch web apps and plugins, and prevent LFI.
- Make the web root read-only and store uploads outside it, so a script cannot be written or executed there.
- File integrity monitoring on web directories to catch new or changed scripts.
- Detect anomalies: scripts in upload folders, web processes spawning shells, odd outbound traffic.
- Run the web server with least privilege to limit a shell's reach.

SOURCES

[1] MITRE ATT&CK: Server Software Component (T1505)

[2] NIST SP 800-83 Malware Incident Prevention and Handling

[3] MITRE ATT&CK: Persistence (TA0003)

Find the backdoors an attacker would leave behind.

securelayer7.net/learn/persistence/what-is-a-web-shell

Open online