Why certain PHP functions are disabled by default

PHP · https://kyzoe.be/en/docs/why-certain-php-functions-are-disabled-by-default/

At **Kyzoe Hosting**, the security, stability and reliability of our hosting environment are paramount. To prevent abuse, hacks and unwanted server load, a number of **risky PHP functions are disabled by default** on our shared and managed hosting platforms.

This measure is in line with best practices in professional hosting environments and meets the requirements for **security hardening**.

Which PHP functions are disabled?

The following functions are disabled by default:

```
exec, system, passthru, shell_exec, dl, popen, show_source,
posix_kill, posix_mkfifo, posix_getpwuid, posix_setpgid,
posix_setsid, posix_setuid, posix_setgid, posix_seteuid,
posix_setegid, posix_uname,
pcntl_exec, expect_popen
```

Why are these functions disabled?

1. Execution of server commands (critical security risk)

Functions such as `exec`, `system`, `shell_exec`, `passthru`, `popen`, `pcntl_exec` and `expect_popen` allow PHP to **execute system commands directly**.

➡ Risk:
With a vulnerable plugin, theme or poorly written code, an attacker can:

- gain shell access
- install malware

- compromise other websites on the same server

---

2. Manipulation of system processes and users

The `posix_*` functions give access to **low-level Unix functionality**, such as:

- user and group permissions
- process management
- system information

➡ **Risk**:
These functions are **not needed for standard web applications** (such as WordPress, Joomla or Laravel), but they can be abused for privilege escalation.

---

3. Dynamic loading of PHP extensions (`dl()`)

The `dl()` function makes it possible to load PHP extensions **at runtime**.

➡ **Risk**:

- Bypassing the server configuration
- Unpredictable behaviour and instability
- Potential execution of unwanted code

---

4. Information leaks via source code

The `show_source()` function can expose PHP source code.

➡ **Risk**:

- Exposure of API keys
- Revealing database details or internal logic

Does my website need these functions?

In **99% of cases: no**.

Popular CMS systems and frameworks work **perfectly without these functions**:

- WordPress
- WooCommerce
- Drupal
- Joomla
- Laravel
- Symfony

If an application does depend on such functions, this often indicates:

- outdated software
- insecure custom code
- an application that would be better suited to a **VPS or dedicated server**

Can I have these functions enabled?

For security reasons, these functions **cannot be enabled on shared hosting**.

Do you have a legitimate use case?

- Contact our support team
- Together we will look at whether a **VPS solution** or a custom server configuration is appropriate

---

In summary

✔ Increased security
✔ Protection against hacks and malware
✔ Better server stability
✔ In line with professional hosting standards

By disabling these PHP functions, we ensure that **your website and other customers stay safe**.