

Forum:[Hack Forums](#)

Thread: [How to detect webshells](#)

Bshol - August 6, 2025, 1:50 pm

Let me know the ways to detect the webshell files in my vps server given that my website is built in PHP. For following options, which one is the best tool to detect the shell files?

- WordPress.
- Other CMS such as Joomla and Drupal.
- PHP frameworks such as Laravel, CI, and so on.
- Pure PHP code.

Thanks for your attention.

itcorp - August 6, 2025, 2:09 pm

(Aug 6, 2025 01:50 PM)Bshol Wrote: [\[CITE\]](#) Let me know the ways to detect the webshell files in my vps server given that my website is built in PHP. For following options, which one is the best tool to detect the shell files?

- WordPress.
- Other CMS such as Joomla and Drupal.
- PHP frameworks such as Laravel, CI, and so on.
- Pure PHP code.

Thanks for your attention.

For wordpress you can take a look Wordfence, right now it might be the best for wordpress. Scans for malware, backdoors, and suspicious files. But there are other scanners similar to it, but when it comes to WordPress your main focus should be on plugins because that's actually its biggest vulnerability... outdated and poorly configured plugins.

DramaKing - August 6, 2025, 2:52 pm

(Aug 6, 2025 01:50 PM)Bshol Wrote: [\[CITE\]](#) Let me know the ways to detect the webshell files in my vps server given that my website is built in PHP. For following options, which one is the best tool to detect the shell files?

- WordPress.
- Other CMS such as Joomla and Drupal.
- PHP frameworks such as Laravel, CI, and so on.
- Pure PHP code.

Thanks for your attention.

Actually, none of those are for detecting malicious code. A quick vulnerability scan and a WPscan check if you use WordPress are all you need other than checking for files that shouldn't be there. A WAF would be overkill, but it will stop webshell activity.

itcorp - August 6, 2025, 3:41 pm

(Aug 6, 2025 02:52 PM)DramaKing Wrote: [\[CITE\]](#) Actually, none of those are for detecting malicious code. A quick vulnerability scan and a WPscan check if you use WordPress are all you need other than checking for files that shouldn't be there. A WAF would be overkill, but it will stop webshell activity.

I get where you're coming from, but just to clarify when you say: "None of those are for detecting malicious code."

That's not entirely accurate. Tools like Wordfence are built specifically to detect malicious code, including webshells, backdoors, obfuscated functions (eval, base64_decode, etc.), and unauthorized file changes. It's not just a vulnerability scanner it actively scans your files, themes, and plugins for known malware and suspicious patterns.

WPScan is great for enumerating vulnerabilities in plugins and themes, and I actually agree with you there, but I already mentioned plugins in my original reply as the main source of risk in WordPress setups. That's why I brought up Wordfence in the first place it complements tools like WPScan by actually scanning the filesystem for infected or altered files.

So again, I totally get the bigger picture you're pointing out (especially when it comes to custom PHP), but since the question specifically included WordPress, I just gave a concrete and reliable solution that works well in that environment. No tool is perfect, but some are definitely better suited to the job than others, depending on the context.

DramaKing - August 6, 2025, 3:47 pm

(Aug 6, 2025 03:41 PM)itcorp Wrote: [\[CITE\]](#) I get where you're coming from, but just to clarify when you say: "None of those are for detecting malicious code."

That's not entirely accurate. Tools like Wordfence are built specifically to detect malicious code, including webshells, backdoors, obfuscated functions (eval, base64_decode, etc.), and unauthorized file changes. It's not just a vulnerability scanner it actively scans your files, themes, and plugins for known malware and suspicious patterns.

WPScan is great for enumerating vulnerabilities in plugins and themes, and I actually agree with you there, but I already mentioned plugins in my original reply as the main source of risk in WordPress setups. That's why I brought up Wordfence in the first place it complements tools like WPScan by actually scanning the filesystem for infected or altered files.

So again, I totally get the bigger picture you're pointing out (especially when it comes to custom PHP), but since the question specifically included WordPress, I just gave a concrete and reliable solution that works well in that environment. No tool is perfect, but some are definitely better suited to the job than others, depending on the context.

Sorry for bring particular, but the question referenced WordPress, not tools built for WordPress. Maybe that was the intent, but it wasn't the wording.

itcorp - August 6, 2025, 3:54 pm

(Aug 6, 2025 03:47 PM)DramaKing Wrote: [\[CITE\]](#) Sorry for bring particular, but the question referenced WordPress, not tools built for WordPress. Maybe that was the intent, but it wasn't the wording.

But to be honest, when someone asks "what's the best tool to detect shell files in my VPS, given that my site is built in WordPress", I think it's pretty reasonable to assume they're looking for practical, WordPress compatible tools, not generic system wide scanners or manual grep work.

That's why I brought up Wordfence because it's one of the few tools that's actually built around the structure of WordPress, with malware scanning, file integrity checks, and specific focus on plugins/themes (which, as I mentioned, are the most common attack vectors) :)

So yeah, maybe the question could've been phrased differently but context matters. If a WordPress user is asking how to detect webshells, I'd rather point them toward something actionable and tailored to their stack, instead of going theoretical.

Either way, all good your points on broader PHP setups are valid too. Different layers of security are always a plus

DramaKing - August 6, 2025, 4:07 pm

(Aug 6, 2025 03:54 PM)itcorp Wrote: [CITE](#) But to be honest, when someone asks "what's the best tool to detect shell files in my VPS, given that my site is built in WordPress", I think it's pretty reasonable to assume they're looking for practical, WordPress compatible tools, not generic system wide scanners or manual grep work.

That's why I brought up Wordfence because it's one of the few tools that's actually built around the structure of WordPress, with malware scanning, file integrity checks, and specific focus on plugins/themes (which, as I mentioned, are the most common attack vectors) :)

So yeah, maybe the question could've been phrased differently but context matters. If a WordPress user is asking how to detect webshells, I'd rather point them toward something actionable and tailored to their stack, instead of going theoretical.

Either way, all good your points on broader PHP setups are valid too. Different layers of security are always a plus

Sometimes I miss things. The question was "For the following options...." You had a good answer.

skull - August 7, 2025, 9:20 am

In most of my websites, I have some automatic warning systems setup.

These are working for not only Wordpress, but any kind of web applications.

I use 2 easy techniques:

- Malware scanning

Im using tools like ClamAV + Maldet to setup automatic malware scanning on the entirety of my servers.

This is fully automatic and can send a report to your inbox with any findings.

Since it is automatic and based of a database, you cannot be sure that this alone will find all shells and malware.

- Monitor diffs

Using GIT for code history changes, allows systems to easily see what files and even what pieces of code has been changed.

This makes it easy to monitor any platform, besides the rules configured in a .gitingore file, so be strict about the .gitignore rules, as a hacker could use these rules to hide stuff in there.

Only use .gitignore for dynamic files, that are secured in a way that no direct file access is available, generally that should be the case for all user uploads. Then even if a hacker changes the .gitignore file, to hide certain other files, the change to .gitignore would still be caught by the monitor for review.

In web application, its most common for hackers to add these as part of existing files, to make it harder to find the change, which is easily monitored with GIT as well, GIT can show exactly what code what changed, not only the files.

While this method is good for manual review, or a AI integration, there are more common practices that

Those are my 2 free go-tos. There are pricey business solutions for this as well, but this is by far the easiest for anyone to do yourself for free.

Bshol - August 13, 2025, 6:48 am

ITCORP, DramaKing,skull.

Thanks for your answers. ?. Sorry about unclear and unspecific situations.

The purpose for my original question was to find ways to detect&remove webshell files on web server assuming my computer was already polluted by hackers.

As for WordPress, Yes, I have some experience in WordFence and know that it blocks various attacks such as sql injection and known plugin weak points, not only detecting Webshells. I agree with ITCORP on WordFence.

The problem is custom php website. I heard that some admins detect the webshell files via admin logs or system logs. Here, I want to clear the meaning of "logs". (Question 1) If the hacker uploaded a shell on the server via the website (f.e. ftp attack), how can I track it? (Question 2) In VPS, or shared hosting server, how can I check the system logs? (Question 3). Which automatic tool or script can I use to find webshell files? (Question 4)

Hope to discuss these matters with you.

Thank you again.