Find Hacks and Backdoors in WordPress & Joomla

Scan your WordPress and Joomla sites for malware, backdoors, and suspicious files. Hash-based detection and 2000+ regex patterns.

Phil E. Taylor | 24 March 2026

The mySites.guru **suspect content tool** scans every file in your webspace for malware, backdoors, and suspicious code - without exception. It's the most-used tool in the mySites.guru audit suite.

If you already know your WordPress site has been hacked or your Joomla site has been hacked, skip straight to our step-by-step recovery guides. Not sure if you've been hacked? Read how to tell if your WordPress site is compromised first. Or if you just want to scan your files, our WordPress malware scanner can check and tell you what's there.

The average number of files across the 80,000+ sites connected to mySites.guru is just under 20,000. The suspect content tool narrows that down to a handful of files worth looking at.

How the mySites.guru audit gathers data

The process starts with a mySites.guru audit. This gathers information on every file in your webspace without exceptions. The audit runs in the background - start it and come back later.

You can schedule audits to run on any frequency, or trigger them on demand. At the start of every audit, we also run the snapshot tools, which add over 100 quick checks on top of the full file scan.

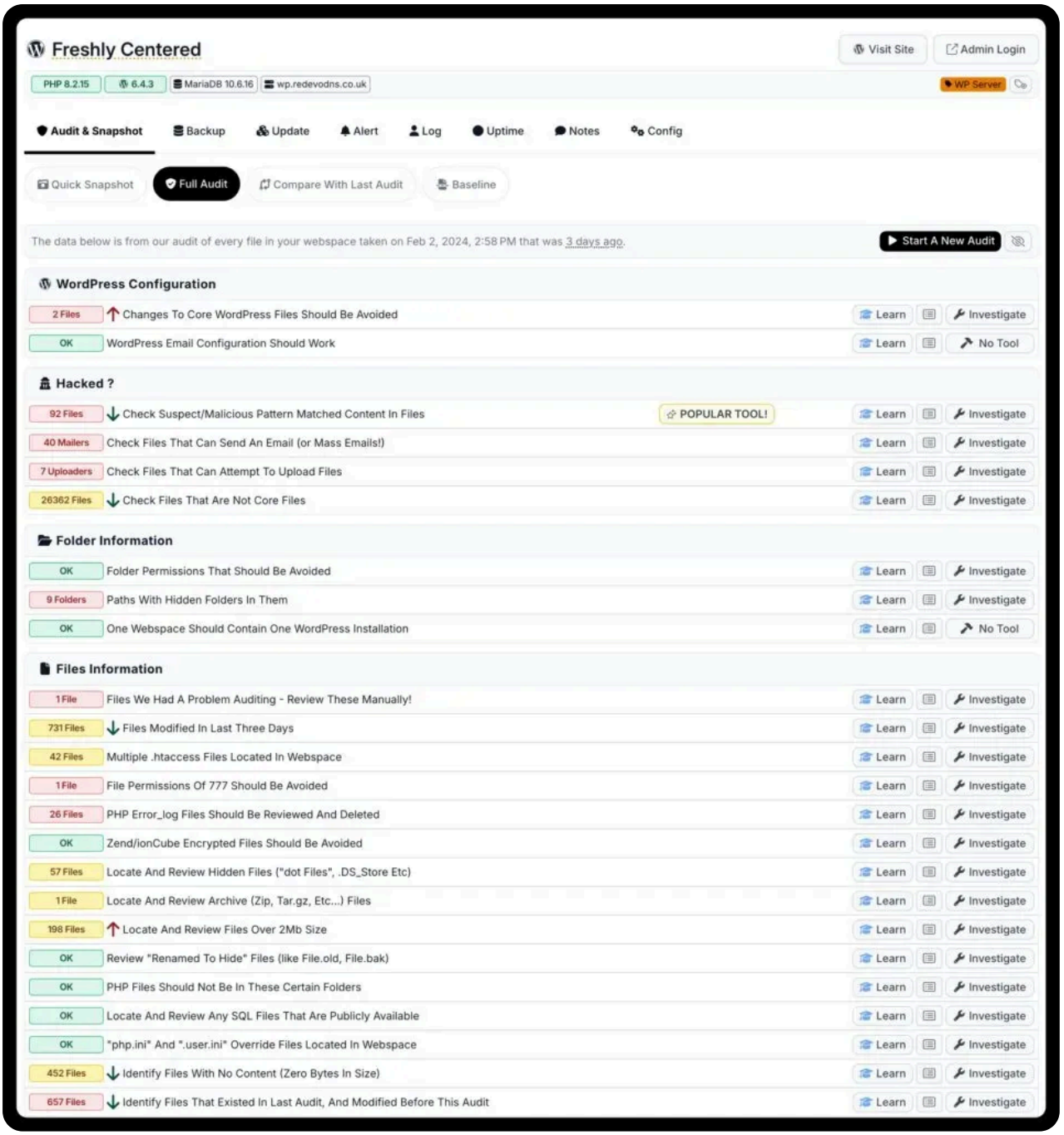
The audit compiles a complete list of every folder in your webspace, then lists every file in those folders.

For each file, the audit:

- Checks whether it belongs to the Joomla or WordPress core
- If it's a core file, verifies it hasn't been modified since release and diffs it against the original

- Saves the md5 hash for future comparisons
- Scans every line against nearly 2000 known hack patterns, labelling matches as “suspect”
- Checks the full file hash against a database of 14,000+ confirmed hacked file hashes - no false positives, each hash is manually validated
- Examines file metadata including creation and modification dates, plus EXIF data on images (a common hiding place)
- Identifies encrypted files, PHP error logs, archives, files over 2MB, zero-byte files, and other anomalous classifications (learn why archive files and SQL dumps are security risks)

When the audit finishes, you get a notification to log in and review the results. The screenshot below shows the first three sections of the audit tab.



Every line in every file

Most “scanners” check the rendered output of your site - what a browser sees. The mySites.guru audit checks the actual files on disk. That includes files not used in rendering at all: dormant backdoors can sit in a webspace for years before a hacker returns to use them. Hackers also use dot-prefixed filenames because most file managers hide them by default - hidden files are a blind spot that deserves its own audit.

Suspect content matching

The audit has two main detection methods:

- **Regex patterns** - over 2000 patterns built from hacks seen on real Joomla and WordPress sites, including recent and mutated variants. A match labels the file as “suspect”. There will be false positives by design.
- **Whole-file hash matching** - a full md5 match against confirmed backdoor hashes marks the file as definitively [HACKED], shown with a red label. These are typically backdoor files we’ve seen before on other sites.

Complete file matching with MD5 hashes

When we find a backdoor (c99, r57, or any confirmed hacked file), we store the md5 hash of the entire file. On the next audit of any site connected to mySites.guru, we check for that hash. A match gets a red [HACKED FILE] flag in the audit results.

There are no false positives here. A hash match is a confirmed hacked file.

04b7a6c9243a604a41230ed34a65f26e	/html.php
38341b81ed79d907c52b31ab682d89f7	/exporter.php
90fb440139127885e80d609d1e20ede8	/libraries/simplepie.lib.php
e90f6b6ed01d2763312bc1e0fcc5c5eb	/simplepie.lib.php
853863ea47d297e20991ce7c101be9a1	/libraries/simplepie/simplepie.lib.php
0067549af911cf2e20de8f409c1f85a3	/cache/cache-db.php
009b1dc7053eeb9845258e1246086002	/libraries/joomla/access/rule.php
b0faf9d72d89f58cc584169525fd32de	/administrator/components/com_joomlaupdate/views/sort-f7.php
393f3f84ad6fb3a0cb85190fa35f3dcf	/pz.htm
4f613ca5170f9c7bba2e7b63ba1624fb	/anon.htm
5ab7216006cf8269307ee158eca749cf	/libraries/simplepie/simplepie.lib.php
e655cd5cf94762b39e14374081d4638b	/cache/cache-db.php

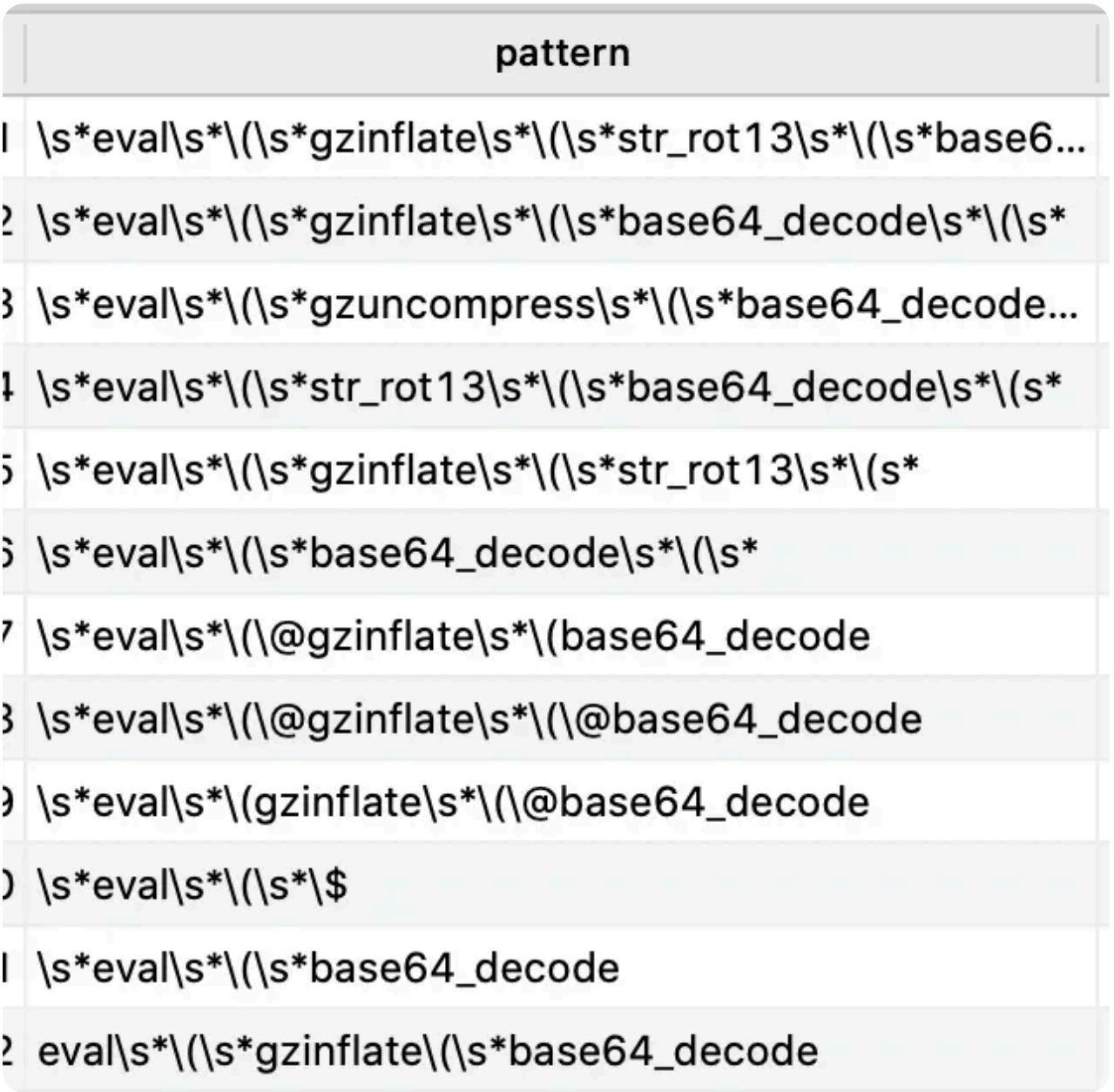
Over 2,000 regex patterns

The second detection layer uses regex: over 2000 patterns built up over a decade, updated daily. They catch common malware signatures like `eval()` combined with

`base64_decode` and `gzinflate`, plus dozens of other patterns.

Regex patterns also find partial hacks - where malicious code has been injected into an otherwise legitimate file, rather than the whole file being a backdoor.

Not every match is a hack, and that's intentional. PHP is used by both legitimate code and by attackers, so some patterns overlap. We work to keep the false positive rate low, but the tool is deliberately inclusive. The result: instead of combing through 20,000 files yourself, you review a handful that the audit flagged.



	pattern
1	\s*eval\s*\(\s*gzinflate\s*\(\s*str_rot13\s*\(\s*base6...
2	\s*eval\s*\(\s*gzinflate\s*\(\s*base64_decode\s*\(\s*
3	\s*eval\s*\(\s*gzuncompress\s*\(\s*base64_decode...
4	\s*eval\s*\(\s*str_rot13\s*\(\s*base64_decode\s*\(s*
5	\s*eval\s*\(\s*gzinflate\s*\(\s*str_rot13\s*\(s*
6	\s*eval\s*\(\s*base64_decode\s*\(\s*
7	\s*eval\s*\(\@gzinflate\s*\(base64_decode
8	\s*eval\s*\(\@gzinflate\s*\(\@base64_decode
9	\s*eval\s*\(gzinflate\s*\(\@base64_decode
10	\s*eval\s*\(\s*\$
11	\s*eval\s*\(\s*base64_decode
12	eval\s*\(\s*gzinflate\(\s*base64_decode

Just a small number of regex patterns we match on

Reducing your time looking for hacks

The average site across the 63,000 connected to mySites.guru has 19,882 files. The audit narrows that down to a short list worth checking, with a built-in interface to view the exact flagged lines - no FTP client needed.

Click any file name to preview the suspect section, along with the file's modification date, size, and permissions. You can edit the file directly in mySites.guru and save it back to the server, or delete the whole file with a single click.

Crowd-sourced data model

After every audit, anonymous data on suspect files goes into a review queue. After manual validation, new patterns and hashes are added to the detection model. This means a hack found on one connected site gets added to the checks run on every other site on the next audit.

It also lets us track waves of infection and detect new and mutated variants earlier.

Detection improves daily

We run over 3000 audits per day, which keeps the detection model current. We find over 200 hacked sites a week. When the Astroid Framework vulnerability was exploited in early March 2026, for example, the suspect content scanner was already flagging the BLPayload backdoor plugins and hacklink cache files across affected sites.

What about false positives?

Not everything that matches our patterns is a hack. This is by design. PHP is used by both legitimate code and by attackers, so some patterns overlap. We keep the false positive rate as low as we can, but we deliberately err on the side of showing you more rather than less.

Can I whitelist files or folders?

No. Whitelisting is not permitted.

You will get false positives, and that's expected. When pattern matching isn't enough to make a call, you can ask me to take a look, or use the AI malware analysis tool to triage suspect files in seconds.

We removed whitelisting after a user whitelisted everything, missed a genuine hack, and sued us. After legal fees we were £14,000 out of pocket. The crowd-sourced data model also means user-supplied whitelists degrade the detection quality for everyone else. I'm the only one who whitelists anything now, and I do it rarely.

Comparison to external scanners

Most services that claim to have an "audit" tool have implemented the Sucuri SiteCheck API, which scans your site as a visiting browser would. It doesn't check the files in your webspace and won't find anything hidden below the rendered output. Not all "audits" are equal.

Current limitations

We don’t currently scan database tables for malware, which means we can miss WordPress SQL-injected posts. That’s on the roadmap.

Out of your depth? Need help?

If the audit finds your site is hacked and you’d rather not deal with it yourself, you can hand it over at fix.mysites.guru for a set-fee hack fix.

Read more in our complete agency security guide.

Frequently Asked Questions

How does mySites.guru distinguish between a definitely hacked file and a suspicious one?

Files matching a known md5 hash from a previously confirmed backdoor are flagged as definitively hacked with no false positives, while files matching regex patterns are labelled suspect and may require manual review.

Why does the suspect content tool produce false positives?

PHP is used by both legitimate code and by hackers, so some patterns overlap; the tool is intentionally inclusive to surface emerging threats rather than miss them.

Can I whitelist files or folders that I know are safe?

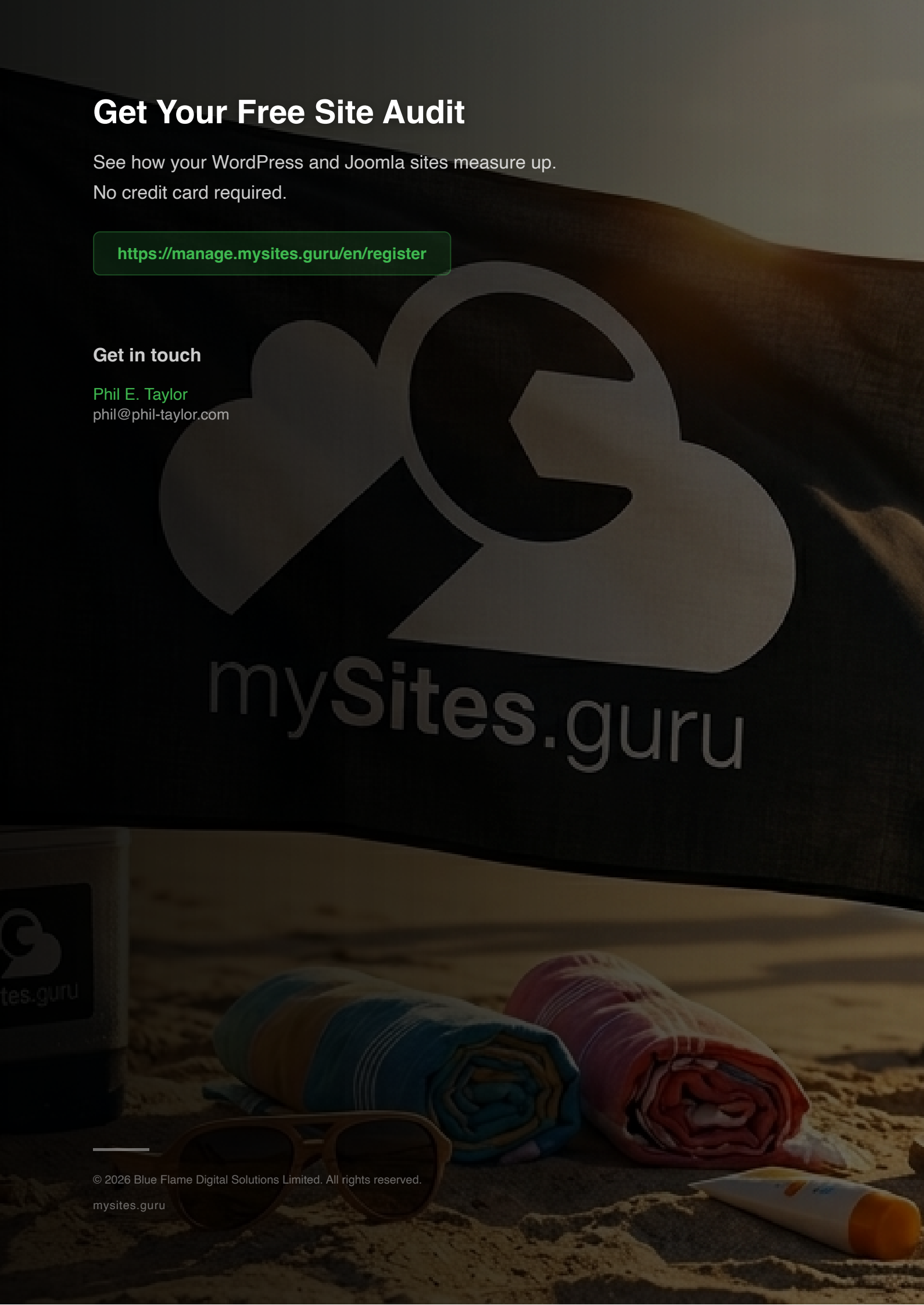
No - whitelisting is not permitted because it reduces the tool's effectiveness and can corrupt the crowd-sourced data model that benefits all connected sites.

Get Your Free Site Audit

See how your WordPress and Joomla sites measure up.

No credit card required.

https://manage.mysites.guru/en/register



Get in touch

Phil E. Taylor
phil@phil-taylor.com