mySites.guru

# DJ-Classifieds Unauthenticated File Upload

DJ-Classifieds below 3.11.2 let anyone upload files to your Joomla site with no login, and it was being used in the wild. Update to 3.11.2 now.

Phil E. Taylor | 20 July 2026

Most of the Joomla extension flaws we write up start with a code audit. This one started with a line in a customer's server log. A site we look after showed an anonymous request hitting DJ-Classifieds' image upload endpoint and dropping a randomly named `.gif` file, with no login behind it. That is the fingerprint of an automated scanner probing an upload it can reach without an account.

DJ-Classifieds is a popular classifieds and ads component for Joomla by DJ-Extensions. We pulled the code apart, reproduced the upload on a controlled test install, and confirmed an unauthenticated arbitrary file upload: the front-end task `index.php?option=com_djclassifieds&task=imageupload` accepts a file with no login and no CSRF token. We reported it privately to DJ-Extensions, and they shipped DJ-Classifieds 3.11.2 as a dedicated security release three days later, on 20 July 2026.

If any of your Joomla sites run DJ-Classifieds, update them to 3.11.2 now, then read on for what the flaw did, why the version's existing "no executable files" defence did not hold, and how to check whether you were already hit.

> **Update to DJ-Classifieds 3.11.2**
>
> 3.11.2 is the fix. It puts authentication and a session token on the imageupload endpoint and restricts uploads to real image types on the server side. Update every DJ-Classifieds site you run. If you cannot update immediately, block requests to the imageupload task at a firewall as a stopgap.

## TL;DR

- **Unauthenticated arbitrary file upload** in DJ-Classifieds, a Joomla classifieds component. The front-end task `option=com_djclassifieds&task=imageupload` writes a file with **no login and no CSRF token**

- We found it from a **customer's server logs**, not a lab: an anonymous request was dropping a randomly named `.gif` through this exact endpoint
- The component's controls stop a bare `.php`, but the malicious-content scan only matches **three literal strings**, so PHP hidden in a valid image with short tags is stored intact
- **Not a clean remote code execution on a default server** (the image is served inert), but a reliable web-shell-in-image plant, and outright **RCE on hosts that run a whitelisted extension** or via a local file include. We rate it **High**
- Affects **every version up to and including 3.11.1** (confirmed live on 3.10.1, by code on 3.11.1, present back to at least 3.9.2). **Fixed in 3.11.2**, released **20 July 2026**
- **No CVE** is assigned for this upload flaw. It is **not** CVE-2025-54474, which is a separate authenticated SQL injection in the same version range
- **Update to 3.11.2 on every affected site, then check for compromise.** DJ-Extensions shipped the fix quickly and published a proper security release for it

## How We Found It, and Why That Matters

The trigger was a single log entry on a site we manage. Stripped down, it was an anonymous request to:

```
option=com_djclassifieds task=imageupload name=<random>.gif filename=<ran
```

No cookie, no session, no referring page from an editing screen. Just a bare hit on an upload task, writing a randomly named GIF. That signature, a scanner probing an upload it can reach without logging in, is worth investigating on its own, and it led straight to the component's upload handler.

This is the part that does not show up in a changelog. DJ-Extensions' own 3.11.2 release notes describe the risk as files being stored "temporarily" and say that "executing arbitrary code was heavily restricted." That is a fair description of the

default-server case, but it undersells what we actually saw: the endpoint was being hit in the wild by an anonymous client before any fix existed. When a flaw is reachable with no login, you do not get to assume nobody has found it. Somebody had.

## How Do I Find Every DJ-Classifieds Site I Manage?

When a flaw drops in a component that could be on any number of your client sites, the first question is always the same: which of my sites run it, and on what version? Logging into forty Joomla admin panels to read the DJ-Classifieds version on each is not a plan when a scanner is already working through upload endpoints.

mySites.guru records the exact version of every installed extension across every connected Joomla site on a twice-daily snapshot. The extension inventory shows every site running DJ-Classifieds, with the version on each, in seconds. Filter for anything below 3.11.2 and you have your work list.

Find every DJ-Classifieds install across your sites

Open your Extension Inventory

Search for DJ-Classifieds across every connected Joomla site and filter for anything below 3.11.2 to find the installs that need updating. Not a subscriber? Sign up free and connect your sites.

Combined with the mass extension updater, you can push 3.11.2 across every affected site in one batch instead of an afternoon of admin logins. That is the difference between patching before the scanner comes back and finding out the hard way.

## What the DJ-Classifieds Upload Flaw Actually Does

The vulnerable code is the `upload()` method in `administrator/components/com_djclassifieds/lib/djupload.php`. The site controller's `imageupload` task calls it directly, with nothing in front of it: no login check, and no Joomla anti-CSRF token. The filename comes from the request ( `name` /

`filename`), and the file body is read straight from `php://input`. The file is written into the component's configured upload directory, which defaults to `tmp/djupload` under your web root and is served over HTTP like any other file.

To be fair to the component, several guards do hold, which is why this is not a one-line `.php` drop:

- **An extension allow-list.** Uploads are limited to a fixed list (`jpg`, `jpeg`, `png`, `bmp`, `gif`, `pdf`, `tif`, `tiff`, `txt`, `csv`, `doc` and similar, through to `zip`). `php` is not on it.
- **A direct block on executable extensions.** A `preg_match('/\.(php|shtml|pht|asp)/i', $filePath)` rejects `.php`, the classic double extension `.php.gif`, and `.phtml`.
- **Filename sanitisation.** `preg_replace('/[^\w\._]+/', '_', $name)` strips path separators, so there is no directory traversal out of the upload folder.

The guard that fails is the malicious-content scan. It reads the uploaded bytes and rejects the file only if it contains one of three literal strings: `<?php`, `eval(`, or `base64`. That list looks reasonable and is trivially incomplete. PHP does not need the string `<?php` to run. A short tag opens a PHP block just as well:

```
<?=system($_GET['c']);?>
```

There is no `<?php`, no `eval(` and no `base64` anywhere in that, so the scan passes it. The older `<script language="php">` form clears it too. Now layer that onto the image requirement. Files with an image extension are validated with `getimagesize()`, and `getimagesize()` is satisfied by a valid header, not a clean file. A GIF that starts with the real `GIF89a` header and carries the PHP after it is a valid image *and* a PHP file at the same time. The allow-list is happy, the extension block is happy, `getimagesize()` is happy, and the three-string scan is happy, so the polyglot is stored on your server byte for byte.

On our controlled test of 3.10.1, that is exactly what happened. A baseline GIF uploaded anonymously and read back over HTTP with a 200. A bare `.php`, a

`.php.gif` and a `.phtml` were all refused with “Wrong file format”. A polyglot GIF carrying `<?=system($_GET['c']);?>` was accepted and stored intact, served back as `image/gif`. There is also a smaller timing issue worth noting: the handler `rename()`s the uploaded temp file into place *before* it runs the extension and content checks and its cleanup `@unlink`, so there is a brief window where the raw file exists on disk under a predictable path.

## Is This Remote Code Execution?

Not on a normal server, and it matters to say so plainly. The whole point of the checks above is that a stored polyglot GIF is served as an image and never executed as PHP. On a default Apache, FrankenPHP or nginx stack, requesting that GIF gives you back an image, not a shell. If a post tells you every unauthenticated upload is an automatic maximum-severity RCE, it is not being straight with you.

What you actually get without any login is still serious:

- **A web shell hidden inside a valid image.** The malicious code sits on your domain, inside a file that passes every image check, waiting for a way to be executed. That is a foothold, and a quiet one.
- **Content and disk abuse.** An open, anonymous upload is a way to host arbitrary files on a trusted domain and to fill the disk.

It crosses into full remote code execution in two common situations. The first is a host that runs a whitelisted extension as PHP, the Apache `AddHandler` or `mod_mime` misconfiguration where `image.gif` or `image.php.gif` is handed to the PHP interpreter. That is not rare on cheap shared hosting. The second is any local file include elsewhere on the site that can be pointed at the stored image; at that point the PHP inside the “image” runs. Either way, an anonymous visitor ends up executing code on your server. That combination, unauthenticated and reachable by anyone, gated on the default stack only by controls that several ordinary configurations undo, is why we assess it as High.

DJ-Extensions' existing controls do reduce persistence: temporary uploads are cleared after roughly twelve hours, which shortens the life of a probe file. That is a genuine mitigating factor. It is not a fix, because twelve hours is plenty of time for a scanner to upload, fetch, and move on.

## DJ-Extensions Shipped the Fix in Three Days

Credit where it is due: this was a fast, clean response. We reported the flaw privately on 17 July 2026, and DJ-Extensions released 3.11.2 on 20 July, a dedicated security release with a public write-up that names the `imageupload` feature and tells administrators to update immediately. That is the right way to handle a report, and it is a good deal better than the silent one-line changelogs we often have to decode.

The 3.11.2 release notes list two changes, and they are the correct two:

- **The `imageupload` endpoint is restricted.** Authentication and CSRF/session validation now stop unauthenticated or unauthorised requests from writing files.
- **A server-side image-only allow-list.** Uploads are limited to real image formats (`jpg`, `jpeg`, `png`, `gif`) on the server, so a `.txt` or any other non-image type is now rejected regardless of what the client sends. The old, broad list that allowed documents, archives and text files is gone.

Those are the same two fixes we recommended in our report: require a logged-in user and a valid token on the upload task, and stop trusting a three-string scan to decide what is safe. So the update genuinely closes the hole.

Two honest caveats for readers. The release notes frame the pre-fix risk as low, leaning on the "no executable files" argument. As the content-scan bypass above shows, that argument had a hole in it, and separately we watched the endpoint being probed in the wild, so treat "update immediately" as the operative line, not "you were probably fine." And the release credits no researcher by name, so if you only skimmed the changelog you might not realise how the flaw was found or that it was already being used. None of that detracts from the fix itself, which is solid and shipped fast.

> **Note: this is not CVE-2025-54474**
>
> There is no CVE assigned for this unauthenticated upload flaw at the time of writing. Do not confuse it with CVE-2025-54474, a separate, authenticated SQL injection in DJ-Classifieds 3.9.2 to 3.10.1 (fixed in 3.10.2, CVSS 8.5) that needs a privileged logged-in user. Different bug, different fix, different risk. The upload flaw is the unauthenticated one, and it is fixed in 3.11.2.

## How Do I Fix My DJ-Classifieds Sites?

Updating is the priority and it is straightforward.

1. **Update to 3.11.2.** In each site's Joomla admin, go to **System** then **Update** then **Extensions**, find DJ-Classifieds, and update it to 3.11.2. If the update does not show there, download the latest build from your DJ-Extensions account and install it over the top. If you run more than a handful of sites, push the update across all of them at once from your mySites.guru dashboard.

2. **If you cannot update straight away, block the endpoint.** Deny anonymous requests to `option=com_djclassifieds&task=imageupload` at a web application firewall or reverse proxy, or block web access to the `tmp/djupload` directory, so the upload never reaches the component. This is a stopgap, not a fix.

3. **Check the upload directory.** Look in `tmp/djupload` (or your configured upload path) for files you did not put there, and for image files whose bytes contain a PHP open tag such as `<?` anywhere inside them. A quick way to spot candidates over SSH:

   ```
   grep -rlie '<?php\|<?=\|<script language="php"' tmp/djupload/
   ```

4. **Check for a break-in, not just the flaw.** Updating stops the next attempt but does nothing about one that already happened. Look for unfamiliar administrator accounts in your Joomla Users list sorted by registration date, and for recently

modified PHP across the whole site. If you find anything, clean the site properly, then change your Joomla passwords and secrets.

Remember that `.htaccess` tricks are a weak answer here. As we covered in why your .htaccess will not stop a Joomla hack, disabling PHP in one folder does not help much when the write happens inside the component and the real risk is an image served from a directory you did not think to lock down.

## Why Image Upload Endpoints Keep Doing This to Joomla Sites

DJ-Classifieds is not an outlier. It joins a steady run of Joomla extension file-upload flaws we have written up, and several share the exact same shape: a front-end endpoint that touches uploads, reachable without a proper login, trusting a check that does not hold. The weakness class is CWE-434, unrestricted upload of a file with a dangerous type, and it turns up across the whole category rather than in one vendor's code.

The recent list makes the point. Page Builder CK (CVE-2026-56290, CVSS 10.0) was an unauthenticated upload that let the attacker pick the destination folder. Balbooa Forms accepted an attachment from anyone with no login and no file-type check. RSFiles and the iCagenda zero-day were both unauthenticated uploads reached without the right checks, and the SP Page Builder zero-day was used to plant hidden Joomla admins. Even jDownloads shipped a stray test upload script with no checks at all. DJ-Classifieds is a variation on the theme: the checks exist, but the content scan trusts a three-string blocklist that a short tag walks straight past.

The thread through all of them is the one we pulled on in AJAX endpoints are a big CMS security blind spot: a front-end endpoint that verifies too little and then does something dangerous. When a framework leaves each extension developer to get authentication, tokens and content validation right on their own, the same mistake gets made independently, over and over. It is also why we are not the only ones auditing Joomla extensions any more, and why this keeps being worth writing about.

## How mySites.guru Keeps You Ahead of the Next One

A file-upload flaw is a race, and the clock starts the moment it is public. The slow part has never been the update itself. It is working out which of your sites even have the affected extension, on what version, and getting the patch onto all of them before a scanner gets there first.

mySites.guru collapses that. Every connected site reports the exact version of every installed extension on a twice-daily snapshot, so "which of my sites run DJ-Classifieds below 3.11.2" is answered the moment you ask it. The mass update tool then pushes the patch to every affected site in one operation. And because this flaw was being probed in the wild, the detection half matters too: the suspect content and hacked-file tool runs on every audit of every connected Joomla site and flags web shells and files that do not belong, including a PHP payload hidden inside an image, without us needing a bespoke signature for this particular attack first. If you are unsure how to read what it raises, we cover suspect content versus a confirmed hacked file separately.



That is the whole point of connecting your sites before the next flaw drops rather than after. When it does, you are not scrambling to build a list. You already have it, you push one update, and you move on. If you are not set up yet, start with a free audit on one site and see your full extension inventory, or sign up for mySites.guru for vulnerability flags and one-click updates across every site you manage. If a site has already been touched, fix.mysites.guru is the done-for-you option.

## Disclosure and Severity

This flaw is CWE-434, unrestricted upload of a file with a dangerous type. It is reachable over the network by an anonymous user, with no login and no user interaction, on any site running DJ-Classifieds up to and including 3.11.1. On a default server the component's controls keep it short of direct code execution, so the immediate impact is a web shell hidden in a valid image plus content and disk abuse; it reaches remote code execution on hosts that run a whitelisted extension or via a local file include. We assess it as High severity. There is no CVE assigned. DJ-Extensions published a security release and shipped the patched 3.11.2 on 20 July 2026.

HIGH
NO CVE

UNAUTHENTICATED

**No login required; fixed in DJ-Classifieds 3.11.2**

Reachable by anyone on the internet with no account, and seen being probed in the wild. On a default server it is a web shell hidden in a valid image plus content and disk abuse; it becomes remote code execution on hosts that run a whitelisted extension or via a local file include. The fix is to update to 3.11.2, or block the imageupload endpoint if you cannot update yet.

No login | Exploited in the wild | Fixed in 3.11.2 | RCE on misconfigured hosts

| Field | Detail |
|---|---|
| Component | DJ-Classifieds (`com_djclassifieds`) |
| Vendor | DJ-Extensions.com |
| Type | Unauthenticated arbitrary file upload |
| Endpoint | `index.php?option=com_djclassifieds&task=imageupload` |
| CWE | CWE-434 (Unrestricted Upload of File with Dangerous Type) |
| Severity | High (our assessment); no CVE assigned |
| Found by | Phil Taylor, mySites.guru, from a customer's server logs |
| Affected versions | Up to and including 3.11.1 (confirmed live on 3.10.1, by code on 3.11.1; present back to at least 3.9.2) |

| Field | Detail |
| --- | --- |
| Fixed in | DJ-Classifieds 3.11.2, released 20 July 2026 |

| Date | Event |
| --- | --- |
| 16 July 2026 | A site we manage shows an anonymous hit on the `imageupload` task in its server logs, dropping a randomly named `.gif`. |
| 17 July 2026 | We reproduce the flaw on a controlled Joomla install, confirm the affected version range by code review, and report it privately to DJ-Extensions. |
| 20 July 2026 | DJ-Extensions releases DJ-Classifieds 3.11.2, a security release that adds authentication and a session token to the `imageupload` endpoint and enforces a server-side image-only allow-list. Update every affected site. |

## Further Reading

- DJ-Classifieds 3.11.2 Security Release - the vendor's own advisory for the patched release, listing the endpoint restriction and the image-only allow-list.
- CWE-434: Unrestricted Upload of File with Dangerous Type - the canonical definition of this weakness class from MITRE.
- OWASP Unrestricted File Upload - why file upload flaws rate so highly and how far they can go.
- OWASP File Upload Cheat Sheet - the developer's checklist for accepting uploads safely, including why a content blocklist is the wrong model.
- Securing Joomla extensions - Joomla's own guidance for developers on access checks and safe file handling.
- A month of Joomla security disclosures - the wider run of Joomla extension flaws this sits alongside.

# Frequently Asked Questions

**What is the DJ-Classifieds imageupload vulnerability?**

DJ-Classifieds is a classifieds and ads component for Joomla by DJ-Extensions. In versions up to and including 3.11.1, its front-end task index.php?option=com_djclassifieds&task=imageupload accepts a file upload with no login and no CSRF token. An anonymous visitor can write an attacker-named file with attacker-controlled contents into the component's upload folder, which is served over the web on a normal install. The fix is DJ-Classifieds 3.11.2, released 20 July 2026, which puts authentication and a session token on the endpoint and restricts uploads to real image types on the server side.

**Which DJ-Classifieds versions are affected?**

We confirmed the flaw live on 3.10.1 and by code review on 3.11.1, and the same upload path is present back to at least 3.9.2. In practice, treat anything below 3.11.2 as affected. The upload helper class was renamed between 3.10.1 and 3.11.1 (DJUploadHelper became DJClasssifiedsUpload, kept as a backward-compatible alias), but every security check in it was identical, so upgrading within the 3.x line before 3.11.2 did not close the hole.

**Is this remote code execution?**

Not on a default server, and we will not pretend otherwise. The component blocks a bare .php upload and refuses obvious double extensions, so on a standard PHP stack a planted image is served inert. It becomes remote code execution on hosts that run a whitelisted extension as PHP (a permissive Apache AddHandler on cheap shared hosting), or when it is chained with a local file include elsewhere on the site. Even where code never runs, it is a reliable way to plant a web shell hidden inside an otherwise valid image, so we rate it High.

**Was this being exploited?**

Yes. We found it because a site we look after showed an anonymous hit on the imageupload task in its server logs, dropping a randomly named .gif with no login behind it. That is what triggered the investigation, so this was being probed in the wild before the vendor shipped a fix, not just a theoretical finding in a lab.

**Is there a CVE for this?**

No CVE is assigned for the unauthenticated upload flaw at the time of writing. Do not confuse it with CVE-2025-54474, which is a separate, authenticated SQL injection in DJ-

Classifieds 3.9.2 to 3.10.1 (fixed in 3.10.2) that needs a privileged logged-in user. This upload flaw is unauthenticated, is the one we saw being used, and is fixed in 3.11.2.

**How do I check whether one of my sites was already used?**

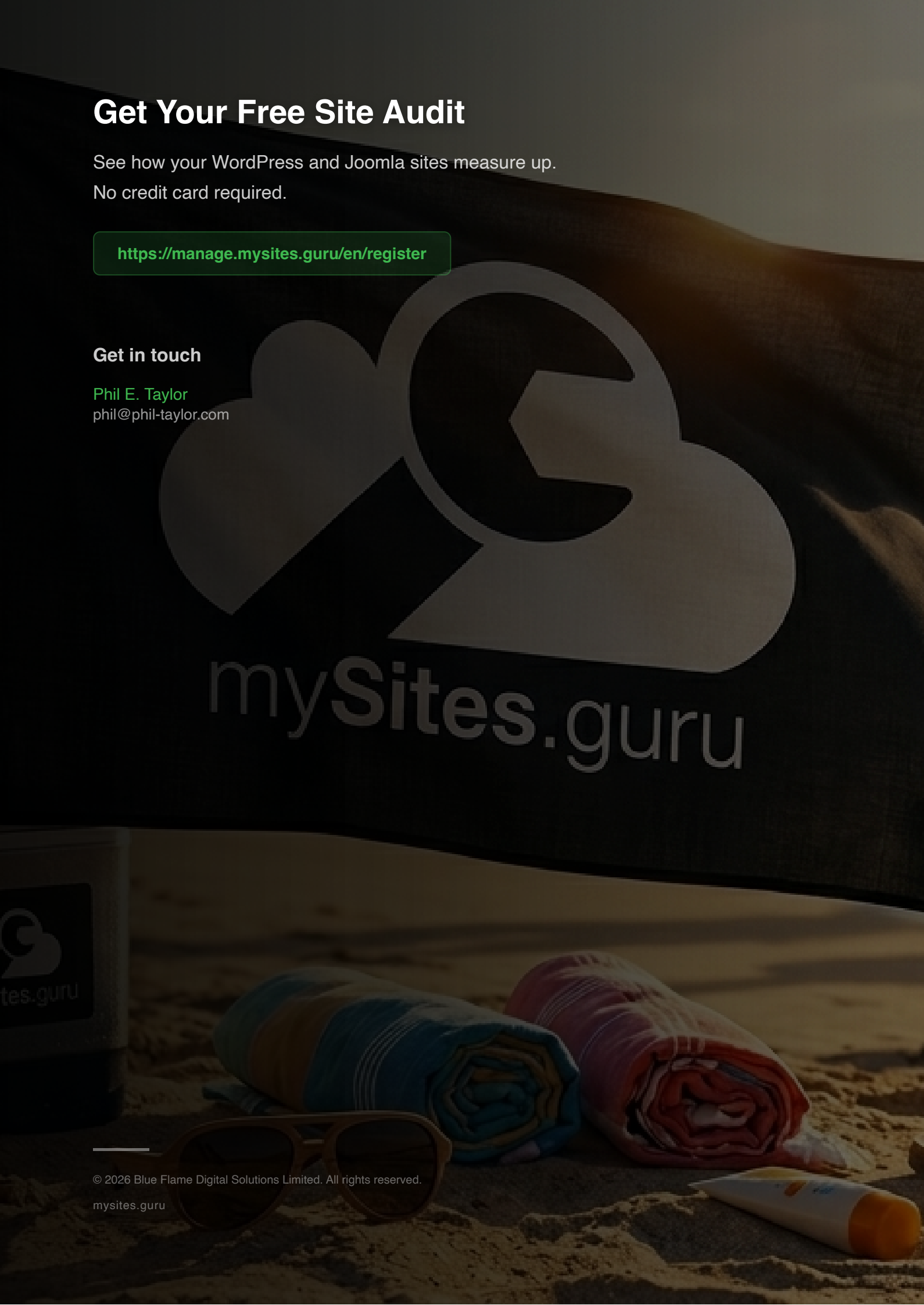
Update to 3.11.2 first, then look for signs of abuse. Check the component's upload directory (by default tmp/djupload under your web root) for files you did not put there, and for image files whose bytes contain a PHP open tag. Look for unfamiliar administrator accounts and recently modified PHP anywhere on the site. On connected sites, the mySites.guru suspect content and hacked-file detection flags web shells and files that do not belong, regardless of which extension let them in.

# 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