Upcoming critical PHP security release (CVE-2026-109618 / -109619) (php.net)

687 points by xd 12 hours ago | hide | past | favorite | 257 comments

https://cert.europa.eu/publications/security-advisories/2026...

https://news.ycombinator.com/item?id=50167754

help

add comment

nr378 14 minutes ago | next [–]

Given that this affects every PHP version from the past decade, and that the exploit instructions are circulating at the top of twitter, and that there's no fix available yet, it doesn't seem outrageous to assume every major PHP website is already compromised?

The upgrade instructions here should include torching your hosts and starting again.

reply

janmo 29 minutes ago | prev | next [–]

Cloudflare: "We are seeing early attacks emerging and are updating mitigations as new vectors show up."

https://x.com/dok2001/status/2112811965735202970

When will they finally release the patch?!

reply

pxtail 12 minutes ago | parent | next [–]

Mindboggling, wtf don't they release it if its ready and exploit is in the wild, its a lot of time until 14:00 UTC and a lot of websites will be breached.

reply

Shank 3 minutes ago | root | parent | next [–]

It's been in the wild for several hours at this point. It was reversed extremely quickly. I would like to hope that someone from the PHP team realizes what kind of Internet-scale disaster they're sitting on and releases it now, but it seems like it should have been released hours ago even.

reply

janmo 5 minutes ago | root | parent | prev | next [–]

They should have shut up, released a new stealth PHP version and told everyone to update ASAP. They gave way to many details ahead allowing some AI agents to find the vulnerability within half an hour...

reply

reddalo 5 minutes ago | root | parent | prev | next [–]

They'll probably release it earlier, that's what they said in the announcement in case the CVE got leaked sooner.

reply

SSLy 26 minutes ago | parent | prev | next [–]

if the attacks are live then PHP should release the fix NOW.

reply

anthonyeden 1 hour ago | prev | next [–]

From what I can tell, Cloudflare has mitigated this via their WAF [0] (free and paid).

The Alpine GitLab[1] indicates the PHP versions to be released 21-10-2026 14:00:00 UTC are:

8.5.12 8.4.27 8.3.36 8.2.35

[0] https://developers.cloudflare.com/waf/change-log/changelog/

[1] https://gitlab.alpinelinux.org/groups/alpine/-/work_items?sh...

reply

edelbitter 45 minutes ago | parent | next [–]

Release notes draft *filenames* (but not contents) are published to the mailing list: https://news-web.php.net/php.webmaster/34116

reply

tro2102 1 hour ago | parent | prev | next [–]

We observed this update ourselves and suspected it to be a mitigation when it was rolled, but I can't find any definitive record that these rules are about the PHP vulnerability. Do you have more info?

reply

anthonyeden 1 hour ago | root | parent | next [–]

https://x.com/dok2001/status/2112802620029223193

reply

ksenzee 1 hour ago | parent | prev | next [–]

I have sites behind Fastly and Akamai, and I would dearly love to hear from either of them.

reply

trog 35 minutes ago | root | parent | next [–]

I have one site on WPEngine and they haven't said anything [that I've seen]

reply

bwoebi 5 hours ago | prev | next [–]

For everyone wondering why the patch is not out yet:

There are a handful people responsible for doing a release. They are still asleep.

To be honest, I am also shocked, that the exploit was published *that* quickly, after the leaked video (which was the reason, why the announcement happened in the first place, at that specific time). We expected a bit more time, even with LLMs...

reply

Shank 0 minutes ago | parent | next [–]

Honestly, kindarails2shell resulted in a responsible disclosure coordination followed by the decision to release early since it was reversed too quickly. Even just a hint of a 0-day is enough to cause people to look for it and exploit it with LLMs.

reply

summner 1 hour ago | parent | prev | next [–]

today only a rumour of a bug is enough to find and replicate it quickly.

anything that narrows the searchspace

reply

jwitthuhn 4 hours ago | parent | prev | next [–]

Hopefully the industry will learn something about how withholding fixes to known vulnerabilities is, in fact, bad policy.

reply

my-huge-pony 47 minutes ago | root | parent | next [–]

How? Biggest organisations and shared hosting providers had a chance to patch and protect their users. Everyone else will learn about the vulnerability the second patch comes out (this is also fixable with a configuration chance, so nobody needs to wait for their distro). This was the right approach and it worked.

reply

jwitthuhn 25 minutes ago | root | parent | next [–]

They didn't release the patch out of fear that someone would quickly use the patch to develop an exploit, meaning an exploit is available shortly after the patch.

What they have achieved by delaying the patch is that exploits are now available publicly before a patch.

Surely this is the worse outcome.

reply

ameliaquining 28 minutes ago | root | parent | prev | next [–]

The problem is that, since proofs-of-concept have leaked but the patch isn't available, bad guys can attack any site whose operators aren't part of the club, and the only thing the operators can really do to prevent that is to take the site offline. (PHP core team has denied that the configuration change fixes the problem.) I would call this a significantly worse outcome than everyone finding out about the vulnerability at the same time, since in that case operators can take timely action if they get the news quickly enough.

reply

heresie-dabord 2 hours ago | root | parent | prev | next [–]

> the industry will learn something

This is something of a fallacy. "The Industry" cannot learn.

But laws and regulations can require better security.

As in the history of seatbelts in vehicles, tech tool stacks are insecure by design and no one is ever really held accountable... until the cost is deemed too high.

It's evident that we have not reached that limit yet.

reply

ftfyyyyyy 2 hours ago | parent | prev | next [–]

https://images.chainguard.dev/directory/image/php looks like patched builds for 8.5-8.2 are available here reply

dlor 2 hours ago | root | parent | next [–]

We've been working with the team on the coordinated disclosure, but I made the call to get these out early when exploits hit in the wild.

The vulnerability and patch are both very complex, and the maintainers did an incredible job on this one. We tested the patch extensively and it's holding up well.

We've made these images free and will keep them that way.

reply

thenaderman 10 minutes ago | root | parent | next [–]

What an incredible irresponsible move essentially 0-daying everyone who was working toward coordinated disclosure to ensure everyone had images or distro packages to update to.

reply

aarondf 2 hours ago | root | parent | prev | next [–]

Sorry the last images I see are from 13 hours ago. Are those for the 2 RCE ones?

reply

thenaderman 30 minutes ago | parent | prev | next [–]

This is nonsense, disappointed to see that publicly from a core dev. Yes, some people are asleep, but the decision was made to wait, to ensure the patch is only made available when users are actually able to install the update and we're giving distros the time to build PHP packages.

reply

baggy_trough 5 hours ago | parent | prev | next [12 more]

Mojah 28 minutes ago | prev | next [–]

I'm keeping a list [1] of all the (possible) precautions one can take, ahead of the patches appearing in a few hours. Exploits are now circulating, according to Cloudflare's CTO, so _any_ mitigation is worth investigating

https://ma.ttias.be/php-cve-2026-109618-early-mitigation/

reply

vwelling 45 minutes ago | prev | next [–]

Node.js has also announced security releases: https://nodejs.org/en/blog/vulnerability/october-2026-securi...

reply

joatmon-snoo 2 minutes ago | parent | next [–]

Unrelated, the Nodejs patch is to strip the emudhra certs, which is its own separate security incident.

reply

tikotus 34 minutes ago | parent | prev | next [–]

Strange coincidence? Can someone see a potential connection here?

reply

CawCawCaw 21 minutes ago | root | parent | next [–]

Could be related to the eMudhra Intermediate CA compromise.

https://emudhra.we-have-a-problem.com/

reply

redox99 6 hours ago | prev | next [–]

The exploit is already out. Yeah this looks like it's going to go on history as the worst vulnerability ever. And now with LLMs you'll likely see tens of thousands of agents hacking any php website out there within hours.

reply

spenvo 5 hours ago | parent | next [–]

Yup, I'm seeing several legit people on X claiming to have seen or found the exploit. Some say that: turning session.upload_progress.enabled off and setting file_uploads = Off will mitigate. But others are claiming that those steps are not enough and "Upload progress can serialize before basic RINT clears BG(serialize)". So, I'm taking no chances and taking my services offline.

https://x.com/MagicalTux/status/2112725774025572431 https://x.com/rina4_tot/status/2112723421641531772

I think this exploit + agentic AI (which allows for rapid secret exfiltration/utilization) may be a potent enough combo to trigger a sort of Kessler Syndrome for network security.

reply

defgeneric 3 hours ago | root | parent | next [–]

If the problem is in the parser then `file_uploads = Off` doesn't help, PHP still parses the multipart body. That seems to be why they're talking about blocking multipart/form-data on routes that don't take uploads and rate-limiting those that do.

reply

int32_64 5 hours ago | root | parent | prev | next [–]

Mark Karpelès? Wow that's a name I haven't thought about since 2014.

reply

TacticalCoder 14 minutes ago | root | parent | next [–]

To be honest that clueless individual and its pathetically lame security practices made him lose 800 000 Bitcoin belonging to its customers with Mt Gox, which was written in... PHP.

And he went to prison for a few years for it. So he knows something about PHP exploits.

reply

matkoniecz 53 minutes ago | root | parent | prev | next [–]

> sort of Kessler Syndrome for network security.

why it would be a thing, beyond "serious problem"?

reply

Narkov 44 minutes ago | parent | prev | next [–]

> Yeah this looks like it's going to go on history as the worst vulnerability ever.

lol no.

I feel old. Heartbleed was 2014.

reply

ameliaquining 9 minutes ago | root | parent | next [–]

Log4Shell was worse than Heartbleed and it doesn't seem totally outrageous to say that this might be worse than Log4Shell.

reply

Fabricio20 1 minute ago | root | parent | next [–]

Log4Shell was on a library called Log4J, used by many (but also not that many: Logback is equally as used and wasn't affected by the same vuln). This is straight up php the language.

reply

bpye 25 minutes ago | root | parent | prev | next [–]

I don't think a PoC was publicly available for Heartbleed before patches? I may be misremembering.

reply

sippingabonedry 55 minutes ago | parent | prev | next [–]

> Yeah this looks like it's going to go on history as the worst vulnerability ever

Yeah no.

This is bad but doesn't even crack the top ten.

reply

dawnerd 4 hours ago | prev | next [–]

I tried it on a personal site, was able to dump the entire database. Immediately shut all my sites down. This is bad.

reply

febed 1 hour ago | parent | next [–]

Does the exploit need a framework like laravel or can it work with just plain php

reply

rossriley 1 hour ago | root | parent | next [–]

No, this happens sometime in the request parsing process, any exposed php URL would be vulnerable.

reply

burnJS 4 hours ago | parent | prev | next [–]

How can I try this to see if I should shut things down myself. I run a 6 figure stealth and have a day job for fintech in 9 figures all php so kinda worried

reply

dawnerd 4 hours ago | root | parent | next [–]

Used this as a base: https://jackalsyn.com/posts/php-up-2026.html

But created my own based on it since it feels sketchy running someone else's exploit poc ya know. Ran it first in a vm before running it against a production site.

reply

burnJS 4 hours ago | root | parent | next [–]

I tried this more simplistic variant https://pastebin.com/kggc4q1T which supposedly would crash php but it was rejected by my CSRF middleware...

reply

afdbcreid 3 hours ago | root | parent | next [–]

If you maintain a 9 figures business, I would recommend to not rely on that and shut it down or at least do more.

reply

system2 1 hour ago | root | parent | next [–]

A 9-figure business with PHP is mind-boggling.

reply

normie3000 1 hour ago | root | parent | next [–]

Wait til you hear about The Facebook.

reply

dan15 35 minutes ago | root | parent | next [–]

Facebook hasn't used PHP for a very long time. It's been using Hack for over 12 years, which was initially based off PHP but it's deviated quite significantly now. It's now nearly entirely statically-typed, and has a bunch of features that PHP doesn't have. IO is fully asynchronous using async/await syntax (like C# or JavaScript), functions can be marked as pure, and a bunch of other things useful in large codebases.

It still uses the same request model as PHP, since that's genuinely useful for web requests. It's easier to avoid memory leaks if each request is entirely self-contained and there's no global state persisted between requests (other than APC and opcode caches)

reply

sznio 3 hours ago | root | parent | prev | next [–]

then you need to use a correct csrf token. it doesn't protect against this, anyone can get a valid csrf token, every visitor gets one.

reply

brazzy 55 minutes ago | root | parent | prev | next [–]

This link gets me a shitton of extremely intrusiv ads, some of which look like malware (auto-downloading an "OperaSetup.exe"). Is pastebin already compromised?

reply

tikotus 36 minutes ago | root | parent | next [–]

I got this same earlier, but now it seems to be working. Perhaps just an oddly timed bug with some ads that has nothing to do with the incident?

reply

sdfgsdfgsdsdfgs 3 hours ago | root | parent | prev | next [–]

On which OS and Server did you run the exploit against?

reply

stackghost 4 hours ago | root | parent | prev | next [–]

You make 9 figures at a fintech? Are you hiring?

reply

burnJS 4 hours ago | root | parent | next [–]

Fintech makes 9 figures. I make 6 there and 6 at my stealth :-)

reply

00deadbeef 3 hours ago | root | parent | next [–]

What is a "stealth"? A secret side project?

reply

rokkamokka 2 hours ago | root | parent | next [–]

Basically taking two full-time jobs and lying about it

reply

theteapot 4 hours ago | root | parent | prev | next [–]

You use PHP at a fintech?

reply

burnJS 4 hours ago | root | parent | next [–]

One of many languages. It has its role.

reply

lolakutty 3 hours ago | root | parent | next [–]

>It has its role...

Of being a hole...

reply

winternett 2 hours ago | parent | prev | next [–]

Meh, I'll just roll PROD back to our 4.7 image, then head down to the pub for a pint until this all blows over... Only 3 years of data will be missing... No biggie! LOL :P

reply

Gigachad 9 hours ago | prev | next [–]

The demo video has been going around Twitter today and apparently the proof of concept is privately being shared around too. The video shows pasting something in the URL and then gaining a shell on the server. It looks completely catastrophic.

I think the best course of action is to just shut your server down before the patch goes live and then bring it back after you have patched.

reply

archon810 8 hours ago | parent | next [–]

https://x.com/k_firsov/status/2112641845926989944

reply

ndiddy 5 hours ago | parent | prev | next [–]

The URL is to show the effects of the exploit. The person's running a python script that runs the exploit and then makes the PHP installation execute an arbitrary shell command. The proof of concept involves writing "pwned" to a publicly accessible location on the web server, then they refresh the browser pointing at the URL to show it's been written.

reply

philipwhiuk 7 hours ago | parent | prev | next [–]

This is gonna rinse so many LAMP WordPress installs where the PHP version is either not getting a patch or the hosting is managed by the user via a contractor or the hosting is managed by a company with a fleet of custom servers etc etc.

reply

bigbuppo 1 hour ago | root | parent | next [–]

It's sad how many wordpress installs haven't been patched in a decade. they're so old even new exploits aren't supported.

reply

dylan604 27 minutes ago | root | parent | next [–]

ha! Who knew ignoring critical updates and continuing to run deprecated software while kicking the can down the road while focusing solely on building new features would turn out to be the best security policy after all.

reply

qingcharles 4 hours ago | root | parent | prev | next [–]

I can think of some really ancient phpBB boards that probably won't ever come back if they get hit.

reply

Gigachad 6 hours ago | root | parent | prev | next [–]

We almost need some bot that logs in to every vulnerable PHP server and turns it off to prevent it being properly exploited before the server admin updates it..

reply

lioeters 6 hours ago | root | parent | next [–]

Kindware instead of malware. Just goes around finding vulnerabilities and using them to patch things and improve security everywhere.

reply

voxadam 5 hours ago | root | parent | next [–]

Benevolent worms.

https://www.schneier.com/blog/archives/2005/12/benevolent_wo...

reply

lioeters 5 hours ago | root | parent | next [–]

Heheh that's cute.

reply

afdbcreid 3 hours ago | root | parent | prev | next [–]

Nice idea, but I don't think you would want e.g. your hospital to have their systems suddenly turn off.

reply

simondotau 2 hours ago | root | parent | next [–]

Absolutely they do. Off today is better than a ransomware attack tommorrow.

reply

pixl97 2 hours ago | root | parent | prev | next [–]

Turning off is better than silent data manipulation by an attacker.

reply

afdbcreid 2 hours ago | root | parent | next [–]

But not than assessment and controlled shutdown or access minimization. Turning off critical systems can result in deaths.

reply

jamesknelson 1 hour ago | root | parent | next [–]

This feels like it might be the trolley problem in real life.

Turning it off may cause deaths. Letting it get ransomwared may cause more. Do you pull the lever to turn it off?

reply

hparadiz 54 minutes ago | root | parent | next [–]

I can't believe I have to write this out in this thread of all places but there is a way to do this.

You make sure www-data CANNOT write ANYWHERE on disk. Not /tmp/ Not to /var/www/. No where.

Then make sure your sudoers has nothing weird in it.

Finally disable www-data's ability to run basically anything that isn't required for the operation of your httpd or php-fpm et al.

You then harden it further and further until an RCE does not matter because it's incapable of doing anything.

tldr always treat httpd as untrusted.

reply

bpye 21 minutes ago | root | parent | next [–]

R/w access to your production database is still a problem, though yes less bad than world read/write.

And also relies on the target being patched against any Linux EoP...

reply

trog 31 minutes ago | root | parent | prev | next [–]

But then WordPress stops working :)

reply

hparadiz 26 minutes ago | root | parent | next [–]

They were talking about critical systems :)

reply

trog 19 minutes ago | root | parent | next [–]

Heh, but for better or for worse, a bunch of people have built businesses out of WordPress and its ecosystem. One of them is my aunt, which is why (as a designated family tech support person, though she does have a part time developer) I'm watching this thread closely!

S0y 4 hours ago | root | parent | prev | next [–]

If people wait too long you could even have the bot use the exploit to update the PHP version.

reply

stackghost 5 hours ago | root | parent | prev | next [–]

I’ve thought about doing this in the past but have no trust in police to see it as a Good Samaritan act

reply

joshuat 6 hours ago | prev | next [–]

If the rumors for these exploits being shared on Twitter are correct this is going to be a nightmare. I wouldn't be surprised if VPS hosts step in to mitigate it somehow since so many cheap/unmaintained LAMP installs are running on those servers.

reply

vincent_s 3 hours ago | prev | next [–]

Two things you can do now:

1. Turn off session upload progress

2. Block multipart requests at the web server

More info: https://www.vincentschmalbach.com/cve-2026-109618-and-cve-20...

reply

Gigachad 3 hours ago | parent | next [–]

If you can afford it, shut down your PHP apps now until the patch is out.

reply

rasz 55 minutes ago | parent | prev | next [–]

>multipart requests

is this multipart request smuggling again?

reply

dubeye 32 minutes ago | prev | next [–]

I had my app running on unsupported php version for 'years', because i didn't have anyone handy to update it.

so even if it takes a few days to get my AI agent to do this, it's still a vastly more secure landscape for an amatuer like me

reply

Gigachad 10 minutes ago | parent | next [–]

You should take your application offline until it’s updated. Otherwise it will be hacked immediately when this exploit comes out.

reply

dubeye 5 minutes ago | root | parent | next [–]

That's not likely to happen. You don't even know what the exploit is.

Anyway my point is I'm much less likely to get hacked in this new era of AI assistance, as I'm only a few days behind, rathe than a few months or years behind.

reply

tikotus 39 minutes ago | prev | next [–]

This has been out for hours! Why is this not the top post here? It's climbing, and hopefully will soon be, but... This is massive, right?

I had to shut down my own backend services. Luckily a big part of my service can run on a static website.

reply

tommica 38 minutes ago | parent | next [–]

It was top post

reply

Mooty 2 hours ago | prev | next [–]

If you use Nginx, you can temporarly block the subject by disallowing multipart sendings to PHP. This is quite a problem for a lot of forms, but way better that being hacked.

Here is an Nginx patch : if ($http_content_type ~* "multipart") { return 403; }

You can also turn of session progress in php.ini as mentionned but that does not seem to cover every cases : session.upload_progress.enabled = Off

reply

trog 2 hours ago | parent | next [–]

Have you tested this against a POC to see if it actually mitigates? Seeing a lot of proposed mitigations but with very little detail about whether they worked or not.

reply

Mooty 2 hours ago | root | parent | next [–]

After digging a little in the PoC code (here https://jackalsyn.com/posts/php-up-2026.html) I'm quite confident it's not a completely working one.

It assumes PHP frees the upload progress data when it reloads the session. But PHP 8.4 (the one we use is 8.4.24, i don't know for other versions) calls Z_TRY_ADDREF(progress->data) to keep that data alive. The fake function call also seems to contain invalid fields.

This makes me question this particular PoC, not the official PHP security warning. I have not reproduced the exploit, so this is a concern based on reading the code, not proof that the real vulnerabilities cannot be exploited.

But anyway the CVE still exists, and the exploitation is tied to multipart, disabling it seems the best thing you can do now without shutting everything down.

reply

nikanj 1 hour ago | root | parent | next [–]

The php devs claim that the only workaround is going to be a new php version https://phpc.social/@Girgias/117476477429221001

reply

sippingabonedry 53 minutes ago | prev | next [–]

Is OpenBSD affected or have all their schizo mitigations finally paid off?

Someone on X mentioned the bug relies on some kind of ASLR bypass.

reply

dangoodmanUT 5 hours ago | prev | next [–]

That is definitely very not good.

I got into software dev because of cyber sec, and almost everything I learned revolved around poorly implemented php, so I always stayed away from it when building

but this time it's php directly? that's scary

reply

mircerlancerous 5 hours ago | parent | next [–]

There are regular security patches for basically everything these days, including OSs. Why does this notice stand out to you? The post suggests that they're being proactive, and discovered and patched these before they were exploited. Rather they take this transparent approach, as opposed to being silent and/or doing nothing.

reply

Gigachad 5 hours ago | root | parent | next [–]

This one stands out because it's trivially exploited over the internet on the default configuration for all PHP versions. Most vulnerabilities require being able to run malicious code on the machine or using some rare configuration.

If the rumors going around are true this will be the worst computer security incident in history.

reply

dgrunwald 1 hour ago | root | parent | next [–]

Are you forgetting about log4shell?

log4shell was easier to exploit. It also come with an initial patch that turned out to be incomplete, so everyone had to urgently patch every java web app twice!

reply

Freak_NL 1 hour ago | root | parent | next [–]

Log4Shell required that the attacker be able to generate a log message containing specific parameters. It was a severe exploit, but not one that worked in every Java application with the affected Log4j versions. Attackers had to find calls that logged user submitted text or query paths first. So the list of prerequisites was larger: had to be using Log4j and had to have logging of user generated text.

Still a very large attack surface, but not nearly as ubiquitous as this one.

reply

afdbcreid 3 hours ago | root | parent | prev | next [–]

It's not "trivially exploitable". Two or three years ago, this would be a critical issue but require quite some expertise to exploit and exploits definitely wouldn't spread like that. LLMs are shaking the entire security industry.

reply

Tadpole9181 2 hours ago | root | parent | next [–]

It's trivially exploitable because it requires no special set up, no special credentials, no privileged access of any kind. If you can call a PHP server and you have the exploit script, you can execute arbitrary code as their `www-data` user.

Usually, attacks are at least conditional and require *something* that weakens security guarantees.

reply

sdfgsdfgsdsdfgs 2 hours ago | root | parent | next [–]

It's not trivially exploitable.

reply

Tadpole9181 1 hour ago | root | parent | next [–]

Thanks, brand spanking new, jumble of letters account!

reply

altmanaltman 2 hours ago | root | parent | prev | next [–]

Yes but that means it is trivially exploitable today since anyone can get access to a LLM and where to point it towards. 2-3 years ago, things were different, that is also correct but that doesn't mean things stay frozen in time.

reply

sdfgsdfgsdsdfgs 2 hours ago | root | parent | next [–]

It's not trivially exploitable.

reply

eithed 1 hour ago | root | parent | prev | next [–]

Heartbleed

reply

enews01 5 hours ago | prev | next [–]

Just shut down our MediaWiki site temporarily out of an abundance of caution. The twitter videos look scary...

reply

1over137 4 hours ago | parent | next [–]

Here's the video (I think): https://nitter.app/k_firsov/status/2112641845926989944#m

reply

throwaway82931 3 hours ago | prev | next [–]

I help manage a few sites hosted on WPEngine, and taking them down is not an easy call to make. After making sure that our environments had the latest everything (WordPress version, PHP version, all plugins and themes on the latest), I contacted WPEngine support briefly.

They're on top of it as I would expect any major PHP hosting service to be. After asking if there was anything I could do to prep for php2shell (seemingly not), I just pleaded to be first in line when they apply patches and let them go.

How are others in similar situations handling this?

reply

aksss 3 hours ago | parent | next [–]

The exploits seem to rely on multipart form submissions. In a normal setup, I don’t believe PHP’s multipart parser processes GET requests.

Therefore, a temp mitigation that runs short of shutting your site down is to temporarily block POST requests until you can get the patches in place. This will screw up any form functionality, including your WP login form, but keeps your site presence up. A failed form submission is probably a better look than no site presence. You can affect this with Apache config or by using a WAF. For instance if you have cloudflare caching your domain, use a free security rule to block all POST requests until your patches are in. Something like `http.request.method in {"POST" "PUT" "PATCH" "DELETE"}`

reply

throwaway82931 1 hour ago | root | parent | next [–]

I've used Cloudflare rules, thanks. I've chosen to be a little more conservative:

```
(any(http.request.headers["content-type"][*] contains "multipart"))
```

reply

lxxvii 21 minutes ago | root | parent | next [–]

Note that Cloudflare rules are case-sensitive. I'm not sure whether PHP's parsing of "multipart" is or not, but you might consider:

```
any(lower(http.request.headers["content-type"][*])[*] contains "multipart")
```

Also, unless your server is specifically set up to only accept requests from Cloudflare, someone could likely send a request directly to your IP, so you might consider blocking multipart requests at the server level. For Nginx I added the following to all /server blocks:

```
if ($http_content_type ~* "multipart") { return 403; }
```

Luckily our service doesn't use multipart for any critical requests.

reply

throwaway800BC 3 hours ago | prev | next [–]

Took a quick peak at the relevant C code and what I think are the offending lines seem somewhat suspicious right off the bat.

Fil-C should have caught this right? I wonder if it would benefit the PHP ecosystem as a whole if the releases are built with it instead.

reply

epse 1 hour ago | parent | next [–]

PHP as it is desperately needed the speedups in the 7 and 8 series releases to be relevant, performance loss of that magnitude is simply not going to be accepted anywhere

reply

sdfgsdfgsdsdfgs 3 hours ago | prev | next [–]

According to this https://jackalsyn.com/posts/php-up-2026.html, you need the dtor to successfully replicate the RCE on a live server.

So, theoretically, if you wanted to hack a server but you don't know what OS/PHP, version they're running, you could compile a list of every delta for every litespeed release and then bruteforce it, right?

reply

herpderperator 1 hour ago | parent | next [–]

So, source-based distros get to shine?

reply

lossolo 3 hours ago | parent | prev | next [–]

Correct.

reply

metax 3 hours ago | prev | next [–]

Dumb question: does this affect a backend even if behind an ouath wall?

reply

jeroenhd 2 hours ago | parent | next [–]

If the backend can be reached, yes.

If you put the oauth wall in front of the application before it can even reach PHP (i.e. in your nginx/caddy/apache config), then no, assuming all paths are covered.

reply

redox99 3 hours ago | parent | prev | next [–]

Probably yes

reply

afdbcreid 3 hours ago | parent | prev | next [–]

If anyone can login, probably yes. If only trusted people can, probably no.

reply

pixl97 2 hours ago | root | parent | next [–]

Just remember that your LLM doesn't count as trusted people.

reply

bakugo 6 hours ago | prev | next [–]

Strange to see this so far down the front page. If the rumors are true, it may be worse than Log4Shell.

reply

ernubufiei 3 hours ago | prev | next [–]

Unsure what the correct response from the PHP team is here - I hope there's been plenty of work done behind the scenes to mitigate this on unpatched servers because there's not a chance defenders win the race everywhere it matters once the patches are out.

reply

redox99 4 hours ago | prev | next [–]

Only 53 comments for what's most likely going to be the biggest vulnerability in history? I don't understand how this isn't getting more attention. The exploit is already out and it's basically shell access on any php website out there.

reply

buzer 4 hours ago | parent | next [–]

It's also very early in EU and evening/night time in part of US. The information that was out earlier was on level "make sure you update tomorrow when the patch comes out", the actual details weren't out yet.

reply

Titan2189 4 hours ago | parent | prev | next [–]

We're all busy turning off uploads on our servers

reply

atymic 6 hours ago | prev | next [–]

Uh oh, POC is already out there https://jackalsyn.com/posts/php-up-2026.html

reply

mschuster91 1 hour ago | parent | next [–]

I haaaaaaate LLM slop. Barely readable due to low contrast and the first sentence already sets off my slop alarm:

> Two cached bucket pointers, one freed tree.

I've rarely been happier than now to have left IT for good...

reply

Perz1val 34 minutes ago | root | parent | next [–]

The writing is what you have a problem with, but the script kiddie h4xxor theme isn't?

reply

nicce 1 hour ago | root | parent | prev | next [–]

I don't know if people understand that this actually lowers their reputation, instead of increasing it. Putting so important text as AI slop. Maybe it is not so important, if not worth of modifying themselves.

reply

meibo 56 minutes ago | root | parent | next [–]

I think the skull already did a good job of that

reply

jacquesm 6 hours ago | prev | next [–]

A lot of people are going to be up late on account of this.

reply

stingraycharles 6 hours ago | parent | next [–]

Must be a pretty serious one that can be exploited easily, I haven’t seen this type of pre-announcement of when something will be released before.

reply

geerlingguy 4 hours ago | root | parent | next [–]

There are some vulns that warrant a pre-release announcement (in the Drupal community, this happened for a few of the more serious fixes). And prior to the public pre-announcement, there is often informal coordination between some of the larger hosting providers or giant companies who can help roll out the patch more quickly.

reply

jeroenhd 2 hours ago | root | parent | prev | next [–]

The proof of concept works against any PHP server (5 and up) running in the default configuration, leading directly to RCE without authentication. The PoC out there can be defeated by disabling multi part uploads.

The PoC also requires a known binary to exploit so custom PHP builds may be safe if you're lucky, but this seems to be an extremely bad one.

reply

Tadpole9181 2 hours ago | root | parent | prev | next [–]

Apparently, any PHP server that accepts a POST request is vulnerable to arbitrary code execution on the host (from the FPM user?) with no defense. This is DEFCON 1 of internet security: 70% of all websites are currently vulnerable until 14:00 UTC tomorrow with zero escape hatch other than blocking traffic.

reply

SyneRyder 6 hours ago | root | parent | prev | next [–]

Check the Twitter linked video above. It's bad. Just having PHP running on the server is enough.

reply

hn_submit 2 hours ago | prev | next [–]

I hope that people will realize that PHP is *unsuited* for public websites and that they need to switch to secure technologies like Java, ASP.NET or Rust.

PHP contains an almost endless stream of vulnerabilities since its inception almost 3 decades ago. Proponents claim that it's getting better but this just goes to show that nothing's changed.

reply

jtchang 3 hours ago | prev | next [–]

This is going to pop a lot of wordpress sites.

reply

system2 3 hours ago | parent | next [–]

Invest in shopify stock this week then. I will tomorrow morning first thing.

reply

system2 2 hours ago | prev | next [–]

God damn Claude (Opus, Fable, either) is not even helping because of their god damn shitty safeguards. Thank you, Anthropic, for letting bad guys win. I can't tell you how much I want open-weight models to work at home that can handle this type of work in the future without bullshitting about safeguards.

reply

t1234s 4 hours ago | prev | next [–]

Would blocking all multipart/formdata requests be a sufficient mitigation until the updated php version is available?

reply

Gigachad 3 hours ago | parent | next [–]

A PHP core dev suggested all the mitigations getting posted are not enough and that if possible you need to shut down any PHP app until the patch comes out https://phpc.social/@Girgias/117476510850918355

reply

adventured 3 hours ago | parent | prev | next [–]

No. Take anything with PHP down until the patch if you can afford to do so.

reply

ipaddr 3 hours ago | parent | prev | next [–]

Yes that's what a lot of people are doing.

reply

minraws 17 minutes ago | prev | next [–]

Well we all got fucked by this cert issue.

reply

alberth 4 hours ago | prev | next [–]

A user on X claims:

```
Temporaly mitigation

php.ini: session.upload_progress.enabled = Off
```

https://x.com/k_firsov/status/2112718728232329240#m

https://xitter.cf/k_firsov/status/2112718728232329240#m

reply

xyst 4 hours ago | parent | next [–]

Doesn’t work

reply

Titan2189 3 hours ago | root | parent | next [–]

Any details why not?

reply

jdiff 3 hours ago | root | parent | next [–]

Core dev says so. So does a release manager.

https://phpc.social/@Girgias/117476510850918355

https://old.reddit.com/r/PHP/comments/1xb2utu/comment/phyu5m...

reply

exodust 2 hours ago | root | parent | next [–]

Off-topic but Reddit now requires sign-in to see old.reddit. They destroyed it, "to keep us safe".

Further more, if in Australia or a few other places like Brazil, UK, France, reddit won't let you sign in at all unless you're proven to be over 16. Bottom line, linking to old.reddit is similar to linking to a paywalled article.

reply

rasz 47 minutes ago | root | parent | next [–]

They degraded old.reddit even for logged in users - images show as <image>and when you click the link you are redirected to new reddit layout just to see the picture

reply

CookieCrisp 3 hours ago | root | parent | prev | next [–]

Several people who work on PHP have said it's not enough, that only the patch will do, but, they're staying silent otherwise

reply

TacticalCoder 7 minutes ago | prev | next [–]

I don't want to jinx it but... A discussion about, of all languages out there, PHP being the one hit by that and not, say, Go or Java or Rust or Haskell etc.?

Can we discuss that at least a tiny bit?

How do people feel about running servers with PHP today?

reply

hparadiz 5 hours ago | prev | next [–]

They need to release it now.

reply

aksss 3 hours ago | prev | next [–]

Rather than taking your site down tonight, if you have cloudflare handling your DNS, orangeclouding your domain, you can use its free security rules to just block all post requests until tomorrow rather than taking your whole site offline. Drop this into a rule: `http.request.method in {"POST" "PUT" "PATCH" "DELETE"}`

Of course you can do this with web server (eg Apache) config as well, but the CF rule is something you can do now on your phone.

I’m not saying I know this works as a mitigation, but it seems like it could at least contribute to a temporary guard until you get patch in place. Everything I’ve seen about the exploits seem to revolve around multipart form submissions.

reply

solyin 3 hours ago | parent | next [–]

You can also do the apache rule from your phone, via ssh, if you're any good.

reply

aksss 3 hours ago | root | parent | next [–]

Haha you’re right, but the skill you need to be good with is probably eye strain tolerance. But must needs and all. :) If you don’t use a WAF of some type, the risk is probably sufficient to warrant firing up the laptop tonight or worst case use the phone.

reply

LoganDark 6 hours ago | prev | next [–]

There is allegedly a temporary mitigation: https://twitter.com/k_firsov/status/2112718728232329240

php.ini:

```
session.upload_progress.enabled = Off
```

reply

robwayne 5 hours ago | parent | next [–]

According to a PHP core dev, this is insufficient, but no details were provided. Probably can't hurt anyway.

https://phpc.social/@Girgias/117476477429221001

reply

lynndotpy 6 hours ago | parent | prev | next [–]

Just a heads up, as of my comment, this is not endorsed by PHP as a migration for the patch.

reply

lol768 5 hours ago | root | parent | next [–]

> as of my comment, this is not endorsed by PHP

Are you speaking officially here?

There are various mitigations suggested online, including completely blocking multipart requests at LB level. Is the combination of both effective?

reply

lynndotpy 5 hours ago | root | parent | next [–]

I am not affiliated with PHP in any way or sense, I am saying this as someone who is able to see PHP's announcements, and there was no patch or mitigation announced by PHP.

I don't think pulling mitigations from random users online during a developing situation is a good idea. I think shutting down ones PHP services for the time being is a very good idea if that's an option.

reply

1over137 4 hours ago | root | parent | prev | next [–]

"LB"?

reply

pickledish 4 hours ago | root | parent | next [–]

Load balancer!

reply

LoganDark 6 hours ago | root | parent | prev | next [–]

I wouldn't skip updating, but is there anything else we can do, other than simply shutting down the web server until the fix is available?

reply

lynndotpy 6 hours ago | root | parent | next [–]

I do not have details about any mitigation. There are rumors that people have independently come across php2shell exploits.

I shut down my servers five hours ago. This might have seemed premature at the time.

I think the best course of advice for most people is to shut down their servers as soon as they possibly can.

reply

qingcharles 4 hours ago | root | parent | next [–]

I applied the mitigation to all my servers when it was released, but now I've stopped all the httpds. A few hours of overnight downtime, vs potentially trying to restore everything from backups tomorrow. Everyone should weigh it up.

reply

xena 6 hours ago | root | parent | prev | next [–]

You could disconnect the web server from the internet!

reply

ThePowerOfFuet 4 hours ago | root | parent | prev | next [–]

The PoC says so.

reply

arjie 1 hour ago | prev | next [–]

Wow, that's a terrible one. Shut down old apps (that I'd forked and upgraded to latest) I had that were PHP. Sadly, my blog was on Mediawiki and so I've got to turn that off too. Tomorrow we learn.

EDIT: Ha, Claude just took the blog and rendered all pages and now it is a static site! I'll restore the Mediawiki once I can fix the issue.

reply

jamesfinlayson 6 hours ago | prev | next [–]

No hints as to what versions are affected?

reply

gnabgib 6 hours ago | parent | next [–]

> These vulnerabilities could affect all PHP versions from PHP 5 onwards

https://cert.europa.eu/publications/security-advisories/2026...

> affect PHP versions 5, 6, 7, and 8.

https://www.bleepingcomputer.com/news/security/php-warns-adm... (which references HN as a source - https://news.ycombinator.com/item?id=50167754)

reply

Operyl 5 hours ago | root | parent | next [–]

I love the "affects PHP... 6" given we never had an official 6 release haha.

reply

gnabgib 5 hours ago | root | parent | next [–]

Yeah.. their text (unedited).. Maybe they meant 5.6? Only 8.x is "supported" currently, though. I wonder how 3&4 avoided it :)

reply

Perz1val 27 minutes ago | root | parent | next [–]

A: I'm writing a post, what versions

B: "5.6, 7, 8“

A: *fixes a "typo"* ok so "5, 6, 7, 8"

reply

cesarb 5 hours ago | root | parent | prev | next [–]

> I wonder how 3&4 avoided it :)

Probably the same way log4j 1.x avoided log4shell: the vulnerable code was introduced on a later release.

reply

simondotau 4 hours ago | prev | next [–]

Once the dust has settled, it would be interesting to give LLMs the pre-patch source code and see whether any of them can identify this chain of vulnerabilities. Even if they can't identify the full chain, can they at least spot one of the constituent bugs and recognise its seriousness?

If they can't, why not?

If they can, why haven't they already?

reply

afdbcreid 3 hours ago | parent | next [–]

If you tell them where the problem is (even rough code area) - I'm 99% sure they can, and have.

If you only tell them to find issues in PHP - the strongest models probably still can, although they might find different issues ;)

reply

marcins 4 hours ago | parent | prev | next [–]

I’d hazard a guess that’s how the bug was found in the first place?

reply

georgemcbay 3 hours ago | root | parent | next [–]

LLMs probably found this multiple times.

The original exploit, and then when the rumors of a big critical patch got out, the people who are leaking the details online probably used an LLM to find the exploit again.

Just revealing that an unpatched exploit exists in $XYZ these days basically guarantees that a bunch of people are going to target $XYZ with LLMs looking for that exploit.

reply

simondotau 3 hours ago | root | parent | next [–]

I would have expected there to be a hundred entities running every single model over major code like PHP, and clang, and pretty much any behemoth open source project the world relies on. This is the sort of thing the model makers should be doing as a matter of course, as part of routine testing of their new models.

reply

SyneRyder 2 hours ago | root | parent | next [–]

Exactly. I thought this is exactly what Anthropic's Project Glasswing was meant to be, and targeted at critical internet infrastructure.

Either Mythos is useless and didn't find this (hint: Opus 5.5 will find this once it knows where to look)... or, they *did* find it, and this is one of many vulnerabilities they're covering up via "guardrails".

I'll be a lot happier if I wake up tomorrow and find this was found with help from Anthropic (or even OpenAI) but I won't hold my breath.

reply

georgemcbay 3 hours ago | root | parent | prev | next [–]

> I would have expected there to be a hundred entities running every single model over major code like PHP, and clang, and pretty much any behemoth open source project the world relies on. This is the sort of thing the model makers should be doing as a matter of course, when they’re testing their new models.

Well, that would be super nice of them.

From a Machiavellian standpoint it may benefit the people with the most powerful models to let some serious RCEs like this PHP one get widely exploited in damaging ways to further their regulatory capture argument that open weight models have to be legally banned and only locked down models with enforced guardrails around usage can be allowed.

reply

simondotau 37 minutes ago | root | parent | next [–]

> From a Machiavellian standpoint

From a marketing standpoint, having the first model to find a bug of this magnitude and severity would be a pretty damn impressive proof of capability. Finding previously unknown, critical vulnerabilities in widely used open-source software could become a gold-standard test of frontier AI competence.

reply

shepherdjerred 4 hours ago | parent | prev | next [–]

I would be astounded if labs were not already fine tuning on known vulnerabilities for their cyber models

reply

simondotau 3 hours ago | root | parent | next [–]

Undoubtedly, but presumably not this specific set of vulnerabilities.

reply

JSalinasAtTER 2 hours ago | prev | next [–]

I'm a sysadmin that manages several servers, I realized I was vulnerable and reproduced what I found on our systems, it appears that simply disabling progress with php_admin_flag session.upload_progress.enabled Off renders the exploit inert...

reply

matkoniecz 32 minutes ago | parent | next [–]

PHP dev and release manager claims differently

reply

aussieguy1234 3 hours ago | prev | next [–]

I suspect we might be about to see a big Bitcoin mining boom...

reply

ronsor 2 hours ago | parent | next [–]

More like Monero, but yes.

reply

charcircuit 5 hours ago | prev | next [–]

How is the patch still not out yet? This is another case of big tech players all colluding together leaving all the small guys out to dry.

At least this time we can have LLMs rediscover it and make patches ourselves.

Here is the patch I had an LLM come up with against master: https://pastebin.com/raw/EUCF0SR0 It at least stops the PoC which is better than nothing.

reply

nullbio 5 hours ago | prev | next [–]

https://x.com/J4ck3LSyN/status/2112703096443097263 Already being exploited. They need to release the patch now, not later. This is already being deployed by attackers as we speak.

reply

dl 3 hours ago | prev | next [–]

I suppose this will go down in history :|

reply

jeffbee 5 hours ago | prev | next [–]

I am seeing a lot of words like "unconditional" thrown around but I remind that defense in depth *is* a thing and you can practice it, or should have been practicing it. You can, for example, disable the `system` function of PHP altogether. You can also arrange for your PHP to be running in a situation where there's no shell, or any other program, to exec. Or for that matter a filesystem.

reply

Operyl 5 hours ago | parent | next [–]

It's a RCE bug, it does not matter if you disable php functions. It's not arbitrary PHP execution, it's an exploit chain resulting in arbitrary code execution.

reply

afdbcreid 5 hours ago | root | parent | next [–]

Indeed, an attacker theoretically can access anything the PHP process has permission to access. However even without limiting permissions (also a good idea) that means the simple/published exploits won't work on your server, and there's a high chance criminals will look for other targets, assuming you're not targeted.

reply

reorder9695 1 hour ago | root | parent | next [–]

Presumably this makes SELinux enabled distros like fedora are significantly less vulnerable, as the PHP process can only access files in a few contexts?

reply

jeffbee 5 hours ago | root | parent | prev | next [–]

You know how "resulting in a shell" works?

reply

bulatb 3 hours ago | root | parent | next [–]

By corrupting memory and causing the CPU to jump from the intended instruction to a different one controlled by the attacker, at which point their instructions control what your process does next. Getting shell does not mean getting it to feed a spicy string to system(), it means tricking something in the process into overwriting memory with bytes that, if read as instructions, tell the CPU to make a system call to spawn a different process running arbitrary code.

You could do that to an internet-connected toaster with a JPEG. The system() function has nothing to do with it.

reply

Operyl 4 hours ago | root | parent | prev | next [–]

I can guarantee you that it's not because someone left system() enabled, it's much lower than that.

reply

jeffbee 3 hours ago | root | parent | next [–]

What is lower than what? This exploit causes the flow of control to reach zend_call_function with the argument "system". It works at the PHP level and having disabled that function is an absolute defense against the exploit.

It isn't 1998 any more and you can no longer just inject shell code on the heap.

reply

Operyl 3 hours ago | root | parent | next [–]

The current PoC in the public gets there via that method, and even then you don't need to call system() alone to cause hell on a host...

reply

jeffbee 3 hours ago | root | parent | next [–]

The point of this thread, and of defense in depth generally, is to limit the scale of "hell" and to trivialize the impact of "on the host".

reply

Operyl 3 hours ago | root | parent | next [–]

Look, at the end of the day, every major security researcher is going "oh shit" and no amount of tweaks to your configuration is going to limit the blast radius in a way that will actually matter. You're going to need to update PHP, that's the _only_ solution to this problem.

reply

calvinmorrison 5 hours ago | parent | prev | next [–]

no idea why this is downvoted. PHP runs typically on linux machines. If you depend on a language feature to prevent 'exploit' and don't utilize the operating systems features to limit access to data.

Shell access is not good, but in many places that wouldn't be the end of the world either.

Rethink your strategy

reply

Operyl 4 hours ago | root | parent | next [–]

If your PHP application accesses a database, you've got exfiltration, it doesn't matter in this case. Unless your PHP application is just echoing non sensitive data (i.e. rendering md files)...

reply

rasz 1 hour ago | prev | next [–]

Shit got real. All my favorite places are hunkering down, like https://www.vogons.org

Due to an upcoming significant vulnerability disclosure with no official mitigation, services are currently offline as a precaution. Services will not be restored until at least later in the day on October 21. This time may also be spent on additional maintenance, so this message may be periodically unavailable.

reply

matkoniecz 31 minutes ago | parent | next [–]

https://community.openstreetmap.org/t/secondary-service-offl...

> The OpenStreetMap.org operations team have decided to temporarily shutdown a number of key secondary OSM.org services due to a critical PHP security vulnerability.

reply

ChrisArchitect 7 hours ago | prev | next [–]

Some more discussion: https://news.ycombinator.com/item?id=50171528

reply

xyst 4 hours ago | prev | next [–]

PHP is truly a garbage ecosystem. I thought log4shell was bad but this is 100X worse.

With vibe coded slop tools readily available, using php in 2026 is just laziness.

reply

codedokode 3 hours ago | prev | next [–]

It is a yet another reminder why we should start new projects with Rust and dig C and C++. Why would anyone want to use C.

reply

barelysapient 5 hours ago | prev | next [–]

This might be the nail in the coffin for php.

reply

mappu 3 hours ago | parent | next [–]

I expect it will be the death of the C codebase which will get an oxidation push.

reply

qwm 18 minutes ago | parent | prev | next [–]

I wish

reply

skoskie 5 hours ago | parent | prev | next [–]

This is a joke, right? I can't tell if this is a joke.

reply

system2 2 hours ago | root | parent | next [–]

It probably is but I am converting everything to python already.

reply

reddalo 51 minutes ago | root | parent | next [–]

...and make no mistakes!

reply

Synthetic7346 1 hour ago | root | parent | prev | next [–]

Not rust?

reply

system2 1 hour ago | root | parent | next [–]

I am used to Python. Hard to switch to something else until I am forced to. I am trying to manage agentic coding at least to the point where I can identify bugs with my own eyes instead of purely believing the LLMs. I know I will cheat and let an LLM take over the entire Rust app if I tried.

reply

aussieguy1234 4 hours ago | parent | prev | next [–]

Given the proven capabilities of LLMs in computer security, get ready for similar issues to affect NodeJS, Java, Go, Rust etc...

reply

mappu 3 hours ago | root | parent | next [–]

NodeJS and Java are in a similar place (this CVE is a memory safety issue in PHP's C-language interpreter) but Go's compiler/runtime is written in Go and is memory-safe all the way down to the assembly.

reply

minitech 1 hour ago | root | parent | next [–]

Much of Node’s standard library is written in JavaScript, which can’t go on to exploit V8 vulnerabilities even given similar types of bugs, so I wouldn’t say it’s in a similar place. Not as familiar with Java, but I expect PHP is in a class of its own here.

reply

aussieguy1234 41 minutes ago | root | parent | prev | next [–]

Looks like we now have a similar release happening for NodeJS

https://nodejs.org/en/blog/vulnerability/october-2026-securi...

reply

minitech 24 minutes ago | root | parent | next [–]

Blog post from a week ago with no “CRITICAL”, any reason to expect it not to look like the last one (https://nodejs.org/en/blog/vulnerability/july-2026-security-...)?

reply

tonymet 5 hours ago | parent | prev | next [–]

Don’t you mean WWZ ? Php is hardly dead – billions of installs.

reply

lolakutty 3 hours ago | parent | prev | next [–]

People ll continue to use it for fear of hurt feelings...

reply

aksss 4 hours ago | parent | prev | next [–]

“PHP is the cockroach of the internet”

reply

shepherdjerred 3 hours ago | prev [–]

Honest question, who uses php anymore?

I know there must be a ton of random sites which makes this a huge deal.

But, does it affect any big players? I think php is big at Meta but I would imagine its so heavily customized it might be safe from this.

reply

garbagepatch 2 hours ago | parent | next [–]

Writers/editors really, really like Wordpress for some reason, which is like 40% of the Internet, add to that all the other Php stuff. At $DAY_JOB we have a Wordpress instance behind our VPN and use the json API to send the articles to a front-end server written in something else. Forces us to duplicate some of the front-end plugin code (and we try to not use many) but we get peace of mind in exchange.

reply

guskel 3 hours ago | parent | prev | next [–]

There's a pretty huge UCaaS software suite that serves millions of seats across the US that runs on PHP 7.4.

reply

dofm 1 hour ago | parent | prev | next [–]

Not really. Just little sites like Wikipedia.

reply

MrVandemar 3 hours ago | parent | prev | next [–]

I use PHP.

Just as a scripting language on my own laptop, nothing prod. But one thing I notice, even coming back to it after a little while, even after using Python for everything, I find I am effortlessly productive with it.

There was an article I read recently about a blind developer who finds it easy to write in, as the "$" for the variable names makes it easier for him to identify what's going on in any given line of code.

Like any language it has advantages and disadvantages.

reply

aksss 2 hours ago | parent | prev | next [–]

I think the ubiquiti edgerouter admin interfaces use PHP. I don’t know about mikrotik, but wouldn’t surprise me if there are a lot of those devices hanging off the internet on some nonstandard port.

reply

bigbuppo 50 minutes ago | root | parent | next [–]

Mikrotik developers would never stoop so low as to use a scripting language written by someone else.

reply

trollbridge 1 hour ago | root | parent | prev | next [–]

Used to. Newer firmware stopped using PHP.

reply

lolakutty 3 hours ago | parent | prev [3 more]

Consider applying for YC's Winter 2027 batch! Applications are open till November 2.