My name is Blake. I am the founder of Building Better Software. I've been writing custom software for 20 years with a passion for building and securing WordPress websites and applications. Keep in mind while you are reading this book that the principles here are well established and apply to any online presence. If you have any questions, feel free to reach out to me at blake@blakehowe.com and I will normally answer within 24 hours.

In the following pages I'm going to walk you through everything you need to know to help secure your WordPress as well as understand the reasoning as to why you could be targeted.

If you feel like this is more than you want to tackle but you want to make sure your website is secure checkout our security and hosting options here

https://buildingbettersoftware.io/services/hardened-wordpress-hosting/
https://buildingbettersoftware.io/services/wordpress-website-care-package/

Why WordPress is targeted

WordPress is by far the most popular website builder in the world, powering over 40% of all websites across the internet. That kind of market dominance makes it an obvious target for attackers. For hackers, the appeal is simple: target the platform that gives them

the highest number of potential victims with the least amount of effort using mostly automated tools.

But the real vulnerability doesn't lie in WordPress core itself at least, not primarily. It lies in the vast ecosystem that surrounds it. This ecosystem is both a blessing (that has contributed to WordPress dominance), but also its biggest weakness

There are many thousands of third-party plugins and themes, many of which are developed by independent creators or small teams. While there are official guidelines and review processes that catch many things, the WordPress project has little direct control over when people update their website, or this third-party code. That creates an environment where insecure or outdated plugins can slip through the cracks and become easy entry points for exploitation on hundreds of thousands of websites on a regular basis.

This isn’t theoretical it’s measurable. In the past year, 47 vulnerabilities were exploited across WordPress core and various themes. That number represents only about 2.5% of all known WordPress related security issues. The remaining 97.5% stemmed from third party plugins and themes. In other words, most WordPress security problems don’t originate in the platform itself they come from the ecosystem built around it.

If you’re interested in how WordPress handles vulnerability disclosures and security responses, Aaron Campbell’s video offers a behind the scenes look at the security team’s process.

There are also active proposals to strengthen security and performance oversight, particularly for third party code. One such proposal can be reviewed here, and ongoing discussions are taking place through initiatives like the WordPress Performance Team.

Meanwhile, security focused platforms like Patchstack are stepping up to fill the gaps. Patch stack offers threat intelligence and vulnerability monitoring specifically for WordPress plugins and themes, helping developers and site owners proactively secure their environments.

Understanding where the real risks lie is the first step in addressing them.

Final Thoughts

WordPress has come a long way from its humble beginnings as a simple blogging platform. It's now the backbone of over 40% of the internet a staggering figure that speaks to its flexibility, community support, and low barrier to entry. But that same openness is a double-edged sword.

The reality is, many WordPress vulnerabilities don't come from the core platform, but from the thousands of third-party plugins and themes that make WordPress so powerful in the first place. This is both its greatest strength and its greatest liability. The core team can only do so much to vet or control that ecosystem. And while processes exist to review submitted code, no system is perfect especially at this scale.

Looking back, the WordPress community has always prioritized freedom and extensibility. That's what made it grow. But looking ahead, security and performance will need to take a more central role. We're entering a time where site speed, data privacy, and

exploit prevention aren’t just "nice to have" they’re expected. Website owners are more aware now. So are hackers.

Take the quiz to make sure you understood this section!

https://buildingbettersoftware.io/security-quizzes/why-is-wordpress-targeted/

Why Would Hackers Target Your Site?

It’s a common misconception that hackers only go after high profile or high traffic sites. The truth is that any website can be a target big or small. Why? Because there’s money to be made, and hackers operate much like businesses: they scale, automate, and seek profit in every opportunity.

Credentials Are a Commodity

There is a thriving underground market for logins, passwords, email addresses, and any personal data they can harvest. With the help of automated tools, hackers can collect thousands of these records in one sweep multiplying their profits much like an e commerce business scale through bulk sales. Even if your site doesn’t store sensitive data, it can still be a gateway.

Hidden Backlinks for SEO Manipulation

One common tactic is inserting hidden backlinks into your site’s content or code. These links boost the search engine rankings of scam or spam websites. Your site continues to appear normal to visitors, but its silently helping shady websites gain visibility and that visibility translates to financial gain for the hackers.

Using Your Server as a Botnet Node

Once compromised, your server can become part of a botnet a network of infected machines used to carry out larger cyber-attacks like DDoS attacks, spamming campaigns, or credential stuffing. This gives hackers more power without needing to pay for infrastructures

Phishing Pages Hosted on Your Site

Some attackers will secretly host fake login pages or phishing forms on your site to trick your visitors into handing over credentials, credit card numbers, or other valuable data. Your site's reputation and trustworthiness help *them* scam users more effectively.

Stealth Is Key

Unlike the early days of hacking where the goal might’ve been to deface or crash a site, today’s cybercriminals aim to stay invisible. They want your site running smoothly because if it goes offline, so does their income stream. Often, malicious actions are targeted

only at specific visitors for example, people arriving from Google search making them even harder to detect.

Final Thoughts

Even if your site is small, hackers can monetize it. They don't need to take over a big brand when they can quietly control thousands of smaller sites and they're counting on you thinking, "Why would anyone target me?"

The better question is: What can I do to protect my site before it becomes part of someone else's business model?

Take the quiz to make sure you understood this section!

https://buildingbettersoftware.io/security-quizzes/why-would-hackers-target-your-site/

Why should you care about keeping your WordPress site safe?

WordPress website security is a critical aspect of maintaining a successful online presence. With cyber threats constantly evolving, ensuring the protection of your website, data, and user information has never been more important. This section will discuss the significance of WordPress website security and the consequences of not prioritizing it.

Protecting Your Data and User Information:

One of the primary reasons to prioritize website security is to safeguard your website's data and user information. An insecure website can be exploited by cybercriminals, leading to unauthorized access, data theft, or manipulation. Ensuring that your website is secure helps protect sensitive information, such as user credentials, customer data, and financial transactions.

Maintaining Your Reputation and Trust:

A secure website fosters trust among your users and customers. If your website is compromised, you risk damaging your brand reputation and losing the confidence of your audience. By investing in website security, you demonstrate your commitment to protecting user data and maintaining a safe online environment.

Ensuring Business Continuity:

Cyberattacks and security breaches can lead to website downtime or data loss, which can be costly and disruptive to your business. By implementing robust security measures, you minimize the risk of downtime, ensuring the uninterrupted

operation of your website and preserving your online revenue streams.

Compliance with Data Protection Regulations:

With the growing emphasis on data privacy, many countries and regions have implemented stringent data protection regulations, such as the GDPR (General Data Protection Regulation) in the European Union. Ensuring your WordPress website's security helps you comply with these regulations, avoiding potential fines and legal consequences.

Mitigating SEO Risks:

Search engines, like Google, consider website security when ranking websites. A compromised website can lead to a drop in search rankings, negatively impacting your visibility and organic traffic. By maintaining a secure website, you can avoid the SEO risks associated with security breaches and safeguard your online presence.

Staying Ahead of Evolving Threats:

Cyber threats are continuously evolving, and new vulnerabilities are discovered regularly. Prioritizing your WordPress website's security enables you to stay ahead of these threats and adapt to new challenges, ensuring the long-term protection of your online assets.

Final Thoughts

The importance of WordPress website security cannot be overstated. By prioritizing the protection of your website, you safeguard your data and user information, maintain trust with your audience, ensure business continuity, comply with data protection

regulations, mitigate SEO risks, and stay ahead of evolving threats.

Take the quiz to make sure you understood this section!

https://buildingbettersoftware.io/why-wordpress-security-matters/

What is Malware and what are some common types

Malware, short for "malicious software," refers to any software specifically designed to harm or exploit any computing device or network. It can include viruses, worms, ransomware, spyware, and many other forms. Malware works by disrupting operations, gathering sensitive information, gaining unauthorized access to system resources, displaying unwanted advertising, and even taking control of affected systems.

Backdoors: This type of malware allows hackers to bypass normal authentication methods and gain remote access to a WordPress site. Once in, they can potentially control everything.

Drive by downloads: These are programs that automatically download harmful software to a visitor's system without their consent or knowledge. It typically exploits a browser, app, or operating system that is out of date or has a security flaw.

Pharma Hacks: This malware injects spam into a WordPress website, usually visible in search engine results but not to the site's users. It's typically used to boost the search engine rankings of illegal pharmacies.

Malicious Redirects: Hackers inject code into the WordPress website that redirects users to a different site. The new site could be a phishing page that gathers user data or a page that initiates a drive by download.

File and Database Injection: These types of attacks can modify or corrupt your website's files and databases, leading to erratic behavior or even data theft. SQL injection is a common type where an attacker can manipulate your website's database, potentially gaining access to sensitive user data.

Final Thoughts

Malware is a major threat to WordPress sites, from hidden backdoors to spam hacks and database injections. These attacks can steal data, ruin your SEO, and damage your reputation fast.

Prevention is key if nothing else take this summary with you:

Keep WordPress, themes & plugins updated

Use only trusted sources

Install a security plugin & firewall

Scan your site regularly

Back up everything often

Take the quiz to make sure you understood this section!

https://buildingbettersoftware.io/security-quizzes/wordpress-malware-awareness/

WordPress Security Threats Beyond Malware

Now we'll dive into key security threats that can affect your WordPress site beyond just malware. Knowing what you're up

against puts you in a better position to protect your site, your data, and your users.

Brute Force Attacks

A brute force attack is basically a digital guessing game except way more aggressive. Hackers use automated tools to fire off thousands of password combinations in hopes of cracking into your site. If you're using weak passwords or the default "admin" account, you're basically holding the door open for them.

SQL Injection

SQL Injection (SQLi) is what happens when your site trusts user input a little *too* much. Hackers can manipulate your site's database queries by injecting malicious SQL code, giving them access to your data or worse, control over your server.

How SQL injection works

Attackers target form fields, URL parameters, cookies, and more anywhere your site accepts input. If you're plugging that input directly into your SQL queries without validation, it's game over. They can read, change, or even delete data.

Cross Site Scripting (XSS)

XSS is the art of sneaky sabotage. Attackers inject malicious JavaScript into your site usually through comment sections or forms and when other users load the page, the script executes in *their* browser. That can mean stolen cookies, hijacked sessions, or malware spread.

Three flavors of XSS:

Stored XSS: Script is saved on your site (like in a comment) and runs whenever someone views it.

Reflected XSS: Script bounces back via URL or form and hits the user immediately.

DOM based XSS: Happens entirely in the browser using existing code no server side changes needed.

Final Thoughts

WordPress is powerful but with great power comes great responsibility (yep, we went there). While malware gets a lot of attention, threats like brute force attacks, SQL injection, and cross site scripting are just as dangerous and often sneakier.

The real secret to security? Stay proactive. Most attacks succeed because someone left a door open whether that's a weak password, outdated plugin, or sloppy code. Update your stuff, use trusted plugins, and keep an eye on your site activity.

Remember, security isn't just about tech it's a mindset. It's about building habits that layer protection over time. Use strong passwords. Enable 2FA. Sanitize your inputs. Monitor your site

Take the quiz to make sure you understood this section!

https://buildingbettersoftware.io/security-quizzes/wordpress-security-threats-beyond-malware/

What are the best Security Practices?

Implementing strong security practices is essential for protecting your WordPress website from potential threats such as hacking, malware, and data breaches. Whether you're managing a blog, business site, or online store, safeguarding your online presence and user data should be a top priority. Below are key security measures every website owner should consider keeping their WordPress site secure.

Keep WordPress Core, Themes, and Plugins Updated

This is by far the most common reason for breeches on a WordPress website. Regular updates are one of the most effective ways to protect your site from vulnerabilities. Developers frequently release updates that include security patches, bug fixes, and new features. Set up automatic updates where possible or routinely check for and install them manually.

Use Strong Passwords and Enable Two-Factor Authentication (2FA)

Implement a strong password policy for all users—including administrators, editors, and contributors. Encourage the use of long, complex, and unique passwords. Enabling two-factor authentication (2FA) adds an extra layer of protection, making it significantly harder for attackers to gain access.

Limit User Access and Permissions

Follow the principle of least privilege: only grant users the access they need to perform their roles. Avoid having multiple administrator accounts and regularly audit user roles to ensure they are appropriate. This limits the risk of unauthorized changes or internal threats.

Perform Regular Backups – offsite and onsite

Schedule regular backups of your entire site, including both files and the database. Store these backups in a secure offsite location (such as a cloud service). In the event of a hack, malware infection, or data loss, a clean backup can save your sites

Use a Web Application Firewall (WAF)

A Web Application Firewall protects your website by filtering malicious traffic and blocking common attacks like SQL injection, cross-site scripting (XSS), and brute-force attempts. WAFs can be cloud-based or integrated via a plugin.

Secure Your WordPress Database

Enhance database security by changing the default table prefix (e.g., from wp_ to something unique), limiting user privileges to only what's necessary, and sanitizing all user inputs. These steps help protect against SQL injection attacks and other forms of database compromise.

Install a Reputable Security Plugin

Use a reliable WordPress security plugin to help manage and monitor your site’s protection. These plugins often include features like malware scanning, firewall configuration, login attempt limits, and file integrity monitoring. Some popular options include Wordfence, Sucuri Security, and iThemes Security.

Regularly Monitor and Scan Your Website

Set up regular scans to detect malware, file changes, and other security issues. Stay alert for suspicious activity such as unexpected login attempts, modified files, or unusual traffic spikes. Many security plugins can automate this process for you.

Use SSL Certificates make sure they auto-renew

An SSL certificate encrypts the data transmitted between your website and visitors, which is especially crucial for sites handling sensitive information like login details or payment data. In addition to security benefits, SSL also boosts your site's SEO and builds trust with users.

Choose a Secure Hosting Provider

Your hosting provider plays a foundational role in your site's security. Choose a host that offers robust protections such as firewalls, malware scanning, automated backups, and DDoS mitigation. Managed WordPress hosting services often include these features out-of-the-box.

Disable Unused Features and APIs

Disable or remove any unused plugins, themes, or APIs (like XML-RPC) that could serve as entry points for attackers. Less code means fewer vulnerabilities.

Final Thoughts

By proactively implementing these best practices, you can significantly reduce your website's exposure to threats and maintain a secure environment for both you and your users. In the following sections, we'll explore each of these measures in greater detail, along with advanced tips and tools to help you build a resilient and secure WordPress site.

Take the quiz to make sure you understood this section!

https://buildingbettersoftware.io/security-quizzes/what-are-the-best-security-practices/

Supercharge Your WordPress Site with Cloudflare WAF

Protecting your website with Cloudflare is one of the smartest, most cost-effective moves you can make—especially for WordPress sites, which are frequent targets of automated attacks. Cloudflare acts as a shield between your site and the rest of the internet, blocking malicious traffic before it ever hits your server. If you're looking for a solid starting point, Troy Glancy's Cloudflare WAF rules are some of the most battle-tested community recommendations around. They cover core protections like

blocking known attack vectors (SQL injection, XSS), filtering bad bots, rate limiting login pages, and securing form submissions

His rules smartly address common WordPress weaknesses: /wp-login.php and /wp-admin/ can be geo-restricted or IP-filtered to limit access to your actual team. Bad bots (like empty user agents or Python scripts) get filtered before they can brute force or scrape. He even includes smart rate limiting for form spam and login abuse—ensuring real users aren't interrupted, but bad actors get locked out.

https://webagencyhero.com/cloudflare-waf-rules-v3/

That said, there are a few enhancements you can layer on top for even more hardened protection:

Directory Traversal Protection: Block patterns like ../ or encoded equivalents (%2e%2e%2f) to stop attackers from trying to reach files outside your site root.

SQLi Pattern Filtering in URLs: Explicitly block URL queries that match patterns like UNION SELECT or contain suspicious id= parameters.

Challenge or Block Non-Standard HTTP Methods: Methods like PUT, DELETE, or OPTIONS are rarely used by WordPress and can be disabled unless you have a specific reason.

Blank or Suspicious User Agent Filtering: Any request without a user agent—or with a suspicious one—can be challenged or dropped. Good bots (like Googlebot) always identify themselves.

Final Thoughts

Securing your WordPress site with Cloudflare isn't just about turning on a few toggle switches—it's about layering smart, contextual rules that evolve with your traffic. Make sure to take full advantage of Cloudflare's expression builder. You can combine threat scores, IP reputation, referrer headers, and even geo-location into precise filters

Keep it dynamic: Regularly review your WAF logs and adjust.

Minimize friction: Start with "Challenge" instead of "Block" where there's uncertainty.

Think like an attacker: Use referrer, threat score, request method, and country together to spot patterns.

Here are a few checklists to help

WAF review

https://docs.buildingbettersoftware.io//Cloudflare_WAF_rules_Checklist.pdf

Cloudflare Security Protocols

https://docs.buildingbettersoftware.io//Cloudflare_Security_Protocols.pdf

Cloudflare Domain Checklist

https://docs.buildingbettersoftware.io//Cloudflare_Domain_Migration_Checklist.pdf

Cloudflare YouTube Videos

What to expect from Cloudflare - https://www.youtube.com/watch?v=U7VW-rM0Sf4

Setup site in Cloudflare - https://www.youtube.com/watch?v=ZEspz38e_f4

A General Tour of Cloudflare - https://www.youtube.com/watch?v=W58yUUl__cI

Setting Up Firewall Rules In Cloudflare - https://www.youtube.com/watch?v=f9qIGD0wJdA

Setting Up And Using Page Rules In Cloudflare - https://www.youtube.com/watch?v=Vddu8LqdacI

Take the quiz to make sure you understood this section!

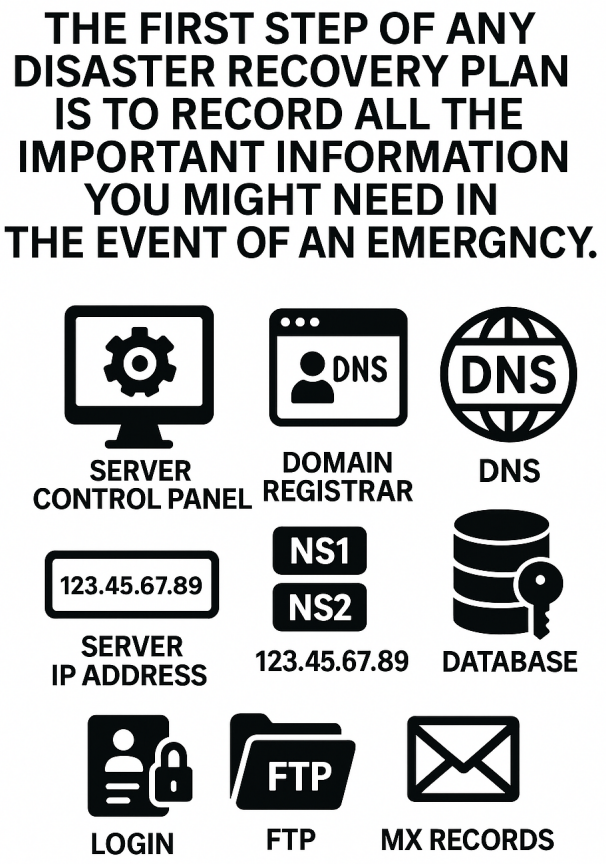
https://buildingbettersoftware.io/cloudflare-for-wordpress/

Implementing a disaster recovery plan

The first step of any disaster recovery plan is to record all the important information you might need in the event of an emergency.

Your Server Control Panel Login Information

Whether you use cPanel, Plesk, ServerCP, or some other control panel, you have to login to access the control panel. Be sure to record not only the login ID and password, but the URL for your login access. Sometimes, this may include a port, so don't forget to include that information as well.



Your Domain Registrar Login Information

If you are moving from one server to another, you will probably need to change your nameservers. To do that, you will need to login to your Domain Registrar's control panel to make those changes, so be sure to record your login ID, your password, and the URL you use to login.

Your DNS Login Information

If you host your own nameservers, you will not need this information. Also, if you use your registrar's name servers, you will not need this information. However, after the recent GoDaddy

debacle, I would recommend against this. If you use an external DNS service, then you should record your login ID, password, and the URL you use to login.

Your Server IP Address

If you host your own nameservers, this will not be important for any of your configuration, but you should still have it recorded. If you use your registrar's nameservers or another nameserver, then you will need the server IP address to record in your "A" record.

Your Name Server Information

You should have at least two nameservers sometimes more. Typically, the protocol would be ns1.domain.com, ns2.domain.com, ns3.domain.com, etc. Some nameservers do not use ns1. but instead use simply ns. Regardless, of the format or how many you have, be sure to record all of them as they are important.

Your Database Information

You will need this for two different purposes.

In case you ever need to reenter it. You will need the database name, the database username, and the database password. Remember when you initially set up your website? If not, set up a test website to see what information is needed to confirm this.

If you use a program to directly access your MySQL database tables. Personally, I use Navicat because I find that it is so much easier to use than PhpMyAdmin. Regardless, of which one you use, you will need to record your database name, database username, and your database password.

Login Information

Be sure to record your main admin login information both username and password. You didn't use "Admin" as your admin login ID did you? If you did, now is your chance to change that. Install UserName Changer from the WordPress repository and make that change NOW, before you take another step.

FTP Information

Next, record your FTP login information. This will include your hostname (domain.com), your username, and password. Also, be sure to note whether you use standard FTP or if you operate through SSH (SFTP) access. Finally, if you use any type of encryption instead of the plain FTP, be sure to make note of that as well.

MX Records

If you are not hosting your own nameservers or if you are using another mail service, you should record your MX Record settings. This would be important to continue to send and receive email should your move your website or you suffer a catastrophe.

Mail Server Information

If you host your own mail server such as Squirrel Mail or some other service then record your online mail server access information. This should include your login ID and password, as well as the URL that you visit to access your email. If you have to designate a specific port to access your mailbox, be sure to record it as well.

Email Provider

If you forward your email to a Google or Gmail account like I do, then you should also record your access information for that account

Take the quiz!

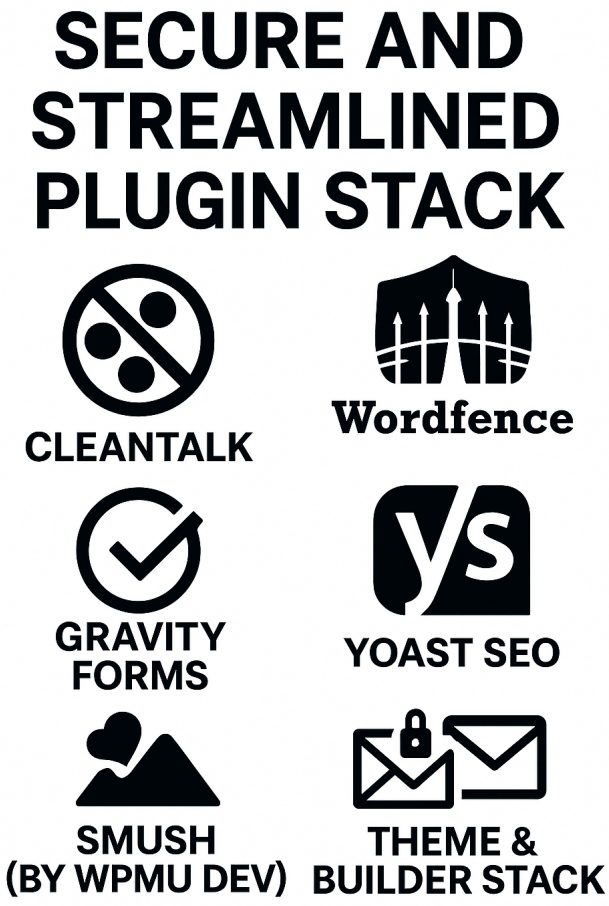
https://buildingbettersoftware.io/disaster-recovery-quiz/

Secure and Streamlined Plugin Stack

Over the years, I've worked on hundreds of WordPress sites from one-pagers to complex eCommerce builds—and in that time, I've refined a plugin stack that balances security, performance, and usability. These tools are what I install on nearly every client site, because they consistently deliver.

CleanTalk

CleanTalk is my go-to anti-spam plugin. It does one job and does it exceptionally well blocking spam in comments, forms, and registrations without the need for captchas. It uses cloud-based filtering and a real-time spam firewall, reducing load on your server while keeping bots out. Spam isn't just annoying it's a gateway for malware, SEO damage, and phishing. CleanTalk keeps those doors shut.



Wordfence

Wordfence is a robust security plugin that offers endpoint firewall protection and malware scanning. It monitors live traffic, blocks malicious IPs, and notifies you of outdated plugins or unexpected changes to core files. Wordfence helps you stay ahead of brute-force attacks, file injection attempts, and exploits giving you visibility into your site's defense.

Gravity Forms

For advanced form building, Gravity Forms is still the gold standard. It integrates cleanly with CRMs, payment gateways, and conditional logic while staying developer-friendly and secure. Secure forms are essential especially contact and lead gen forms.

Gravity Forms helps ensure your user input points aren't weak links.

Yoast SEO

Yoast helps you manage on-page SEO, sitemaps, metadata, and social previews in a structured, user-friendly way. A secure, fast site is great but if it can't be found, it's not doing its job. Yoast supports technical SEO best practices without needing custom code.

Smush (by WPMU DEV)

Smush handles image optimization automatically: compressing images, stripping metadata, and lazy loading where needed. It's simple, efficient, and integrates well with most caching setups. Speed is security. Fast-loading pages discourage bots, help your SEO, and improve user experience.

WP Mail SMTP

WordPress's default email handling is unreliable. WP Mail SMTP fixes that by rerouting email through secure, authenticated SMTP providers like SendGrid, Gmail, or Mailgun.

Why it matters: Whether it's form notifications or password resets, if your emails don't deliver, your site is broken in the eyes of your users. WP Mail SMTP ensures reliability.

Theme and Builder Stack

Depending on the project scope and team skillset, I typically go one of two routes:

A. ACF + Elementor + Hello Elementor

This stack works great for rapid development, design flexibility, and client-friendly editing. Elementor makes it easy to create beautiful front ends, while ACF (Advanced Custom Fields) provides a structured backend. Fast turnaround, visually dynamic pages, or marketing sites that need frequent updates by non-devs.

B. ACF + Gutenberg Blocks + Custom Theme

For sites requiring higher performance and tighter control, I use custom Gutenberg themes with block-based design. ACF bridges dynamic data needs without bloating the frontend. Performance-first builds, content-heavy sites, or when page builders aren't the right fit.

My website has been hacked how can I fix it?

First of all, I hope you too my advice and implemented the diester recovery plan, if so, you're already a step ahead.

Step 1 – Set the site in maintenance Mode

Go Into Maintenance Mode take your site offline temporarily using .htaccess rules or your hosting panel to prevent further damage or visitor exposure.

Step 2 – Scan and Remove Files

Scan for Malware with Word fence. Make sure to enable file change monitoring, firewall, and brute force protection

https://wordpress.org/plugins/wordfence/

Suspicious files (e.g., in /wp-content/uploads/, /wp-includes/, etc.) Should show up in your Word fence scan, you can opt to just let Word fence clean them up for you

Check your functions.php to see if its been modified usually it have code you can't read at the top or bottom of the file remove it but be careful this can easily break your site. If your not sure you can use an LLM like chatgpt.

Remove any Unknown plugins or themes

Replace Core Files – Go ahead and download a fresh copy of WordPress and replace everything except your wp-config.php, wp-content/, and htaccess.

After this is done scan with https://sitecheck.sucuri.net/ this tool is great at catching injected JavaScript if your site still comes back as blacklisted search the database using a tool like phpmyadmin for that string, it's probably embedded in a plugin or post.

Step 3 - Reset and Harden

Reset all Passwords for WordPress admin users

Database credentials (wp-config.php)

Any FTP/SFTP

Hosting control panel

Step 4 - Secure Your Site Going Forward

Update Everything - WordPress core, All plugins & themes. Make sure to remove anything unused or outdated. At this point it's better to temporally break something than leave a potential vulnerability. They will normally show up as "abandoned" in Word Fence

Enable Backups - Set up regular off-site backups there are tons of plugins to do this with. Updraft is a great one https://wordpress.org/plugins/updraftplus/

Staying informed about WordPress

To stay up to date on WordPress news, there are several reliable sources and methods you can utilize:

WordPress Slack

https://make.wordpress.org/chat/

WordPress Forums

https://wordpress.org/support/forums/

Make WordPress

https://make.wordpress.org/

Official WordPress Blog:

https://wordpress.com/blog/

WordPress Community Events:

https://wordpress.org/events/

WordPress News Websites and Blogs

https://www.therepository.email/

Follow official WordPress social media accounts you can find all 11 here

https://make.wordpress.org/marketing/handbook/social-media/

Admin bar Facebook Group

https://www.facebook.com/groups/2147806538801573

Post Status Slack

https://poststatus.com/post-status-on-slack/

Final Thoughts

By leveraging these resources, you can stay informed about WordPress news, updates, security releases, best practices, and community events. Keep exploring the WordPress ecosystem to expand your knowledge and stay ahead in the ever-evolving world of WordPress.

Summary of the Recent WordPress Drama

The recent dispute between Automattic (led by WordPress co-founder Matt Mullenweg) and WP Engine has sparked one of the most public and consequential reckonings in the WordPress ecosystem ever. At the center of this controversy are issues of trademark usage, open-source contributions, governance, and a deeper philosophical divide: the “maker vs. taker” debate.

Mullenweg has accused WP Engine of being a “taker”a company that profits significantly from the WordPress ecosystem without giving enough back to the core project. From his perspective, Automattic and other contributors bear the cost of maintaining and advancing the open-source software, while commercial entities like WP Engine benefit disproportionately. This imbalance, if left unchecked, poses a threat to the long-term health and sustainability of the WordPress project.

WP Engine, on the other hand, rejects this framing. The company argues that it contributes in non-code ways supporting community

events, educating users, and empowering businesses that rely on WordPress. Furthermore, it challenges whether Automattic, as both a commercial player and gatekeeper of the WordPress trademark, can fairly police who is or isn't contributing "enough" without a clear conflict of interest.

A Call for Reform

In the wake of this drama, many in the community are calling for real reform:

Governance Overhaul - There's a push for a more transparent and community-driven model that reduces the dominance of any single company.

Trademark Clarity - Future policies may better distinguish between WordPress as software and commercial uses of the brand, avoiding ambiguity.

Redefining Contribution - Broader definitions of what it means to "contribute" are being explored whether through code, education, infrastructure, or support.

Legal and Ethical Precedents - The outcome of this lawsuit could shape how open-source projects define governance and enforce accountability moving forward.

Here are a few resources to get you up to speed

https://bullenweg.org/
https://gist.github.com/adrienne/aea9dd7ca19c8985157d9c42f7fc225d
https://mullenweg.wtf/
https://antimattic.net/

The Fair Package Manager

The FAIR Package Manager is a new initiative launched under the Linux Foundation to decentralize the way WordPress sites receive plugins, themes, updates, and translations. Traditionally, WordPress relies on a central repository maintained by Automattic via WordPress.org, creating a single point of control and potential failure. The FAIR project addresses this by allowing developers, hosts, and agencies to run their own trusted repositories, enhancing supply chain security and eliminating unnecessary telemetry. It introduces a WordPress drop-in plugin that reroutes package delivery to these federated sources.

This project arose in response to growing concerns within the WordPress ecosystem, particularly after disputes involving Automattic's restrictive update policies toward competitors like WP Engine. By shifting control from a single entity to a Linux Foundation–backed, community-led governance model, FAIR aims to restore balance, improve compliance, and foster innovation. With strong technical leadership and support from key figures in the WordPress community, it represents a structural shift toward transparency, autonomy, and resilience in WordPress distribution

Final Thoughts

This is more than a corporate feud it's a turning point for WordPress. It forces the community to confront uncomfortable questions about who benefits from open source, who carries its burden, and how power should be shared. As the platform continues to scale, it must evolve to balance commercial innovation with the foundational values of openness and collaboration.

Whether through formalized contribution models, clearer trademark rules, or new governance structures, WordPress now faces a critical opportunity: to reassert its identity as a truly open, community-led platform or risk being shaped by the very forces it was created to resist.