WordPress Security Improvements

Content Management System

A Content Management System (CMS) is a software application that helps users create, manage and modify content on a website without the need for specialized technical knowledge. In other words, it helps to build a website without any coding. Some of popular CMS are WordPress, Joomla, Drupal, Magento and Squarespace.

CMS typically has two major components:

1. Content Management Application (CMA) - to add and manage content
2. Content Delivery Application (CDA) - to fetch the content and makes it visible to visitors

The best CMS platform for Websites

WordPress (WP) was introduced in 2003 for building blogs and enabled everybody to contribute to its improvement. Besides blogging, it has extended further with multiple themes, plugins and specialized hosting providers. WP has turned into a complete web solution for e-commerce sites, blogs, news and enterprise-level applications. Its open-source nature and plugin architecture allows developers to use it as an online shop, a photo gallery, a news website or anything.

WordPress became quite popular to build CMS websites, which powers 34% of the web, based on the data from a web technology survey firm W3Techs. According to WordCamp, there are more than 75 million sites that make use of WordPress. At this rapid pace, we can anticipate worldwide usage to reach nine digits in the near future. A report from Google Trends in December 2018 revealed that the term “WordPress” had over 18 million searches.

What is Vulnerability?

Website vulnerability is a weakness in a website that allows a malicious actor to perform any unauthorized action. Most vulnerabilities are exploited through applicable tools or techniques that can link to a system weakness. Some attackers alter system resources or affect their operation while others steal information from the system but do not alter system resources. Few common website vulnerabilities are SQL Injection (SQLi), Cross-Site Scripting (XSS), File Inclusion (LFI/RFI) and Cross-Site Request Forgery (CSRF).

WordPress Vulnerabilities

WP’s security team works hard to keep the platform secure for its end users. It has a team specifically devoted to locating, detecting and fixing security issues that arise in the core code. The fixes are immediately released. WordPress core is secure when we upgrade it on a regular basis.

Even with a secure WordPress core, the end user cannot totally rely on the WP core functionality. Numerous plugins and themes are available for free and pro, which will save development time and the cost. But, installing more themes and plugins can create loopholes to get exploited by hackers. If a plugin has a vulnerable security issue, the whole site may get hacked.

Thousands of WordPress websites are prone to hacking every year. They get hacked more when compared to other CMS like Magento, Joomla, Drupal and Open Cart. Close to 90% of the scanned WP sites were infected with one or more vulnerabilities, as per SUCURI. Among the nearly 4,000 known WP vulnerabilities, about 52% of them are present in WP plugins and 37% are in the WP core, while the rest belong to WP themes.

How to secure our WordPress site?

In spite of the vulnerable nature of WordPress core, we can keep our website secure by testing and fixing. MOURI Tech takes utmost care by performing periodical testing and fixing to keep it secure. Below are some common vulnerabilities and fixes that we do with all of our WordPress websites.

1. **Check for Vulnerability**

Our dedicated team at MOURI Tech makes it a priority to check for vulnerabilities in themes and plugins while selecting the same for our development. We use the plugins and themes which don't have any vulnerability after completing a thorough analysis.

2. **Latest Version**

At MOURI Tech, we always use the latest versions of PHP, MySQL and WordPress plugins and themes because the latest version will always have the fixes for any kind of security vulnerability.

3. **Regular Updates**

One cannot just relax after the site is deployed with a lot of work. More effort is required to maintain it. We continuously check for new releases and upgrade our site immediately to keep it secure.

4. **Complex Credentials**

We make sure to use complex credentials – a mix of lower and upper case letters and special characters for MySQL username & password, WordPress admin username & password and database prefix.

5. **Security Patches**

Apart from the above common practices, MOURI Tech has a special team to test the developed site and report any vulnerabilities. The development team attends to those issues, does research and performs additional patches. Some of the common practices are listed below.

- ***Sensitive Data Exposure***

  Sensitive data exposure occurs when an app does not effectively protect sensitive information. If passwords are exposed, the attacker can abuse these credentials. We add our custom code to encrypt the password before login or register form getting submitted with SHA256 encryption.

- ***Clickjacking***

  Clickjacking occurs when an attacker makes use of several transparent or opaque layers to trick a user into clicking on a button/link on another page when they were planning to click on the top-level page. To avoid this issue, we change the X-Frame-Options to Same Origin using .htaccess file.

- ***Security Misconfiguration***

  As per OWASP, attackers frequently attempt to exploit unpatched flaws or access default accounts, unused pages, unprotected files, directories and so on to gain unauthorized access to the system. To overcome this issue, we remove the PHP version in response header (php.ini), JavaScript and CSS versions (style_loader_src hook) and restrict access to certain JS files which show internal form fields.

- ***Using Known Vulnerable Components***

  This type of threat is encountered when the components like libraries or frameworks used within the application almost always execute with full privileges and this could be fixed by upgrading to the latest PHP version.

- ***Brute Force Attack***

  A brute force attack can occur in many ways, but it predominantly consists of an attacker configuring predetermined values, making requests to a server using those values and analyzing the responses. It is an attempt to find admin password by trying every possible combination of letters (upper case/lower case), numbers and symbols until the one correct combination that works is discovered. We integrate reCAPTCHA with login forms by setting lockout count to 3.

- ***Username Enumeration***

  User enumeration is when a malicious attacker uses brute force to guess or confirm valid users in a system. For this, we change the error message not to expose any clue on the actual error when login attempt is failed.

- ***Dangerous HTTP Methods***

  It is known that PUT, DELETE and TRACE methods are vulnerable. We disable all the http request methods except GET and POST requests in Apache using .htaccess file.

- ***Directory Listing***

  Misconfigured or default web server configurations display the list of files contained in the directory that are not supposed to get exposed as this might aid malicious hackers to craft a hack attack. We change the hosting server folder settings using .htaccess to avoid listing directly.

- ***DoS Attack***

  Denial of Service (DoS) attack is aimed at making a resource (site/application/server) unavailable for the purpose it was designed. We add reCAPTCHA in all the front-end forms.

- ***Insecure SSL***

  Application configuration should ensure that SSL is used for all access-controlled pages. This happens since there are links on the page that still point to HTTP instead of HTTPS. We always make sure every link from our website points to HTTPS only.

- ***XSS***

  According to OWASP, Cross-Site Scripting (XSS) attacks are a type of injection, where malicious scripts are injected into otherwise trusted websites. This enables the attacker to execute malicious scripts, resulting in hijacking user sessions, defacing websites or redirecting the user to untrustworthy sites. We always block special characters and sanitize the text input field in all the front-end forms.

- ***DMARC Vulnerability***

  Missing SPF records is a frequently encountered security issue which puts sensitive information at risk. Spammers can make the "From" address on email to make messages appear to come from our domain and our domain quality is negatively affected. SSL providers provide certificate to overcome this issue.

- ***Malicious File Upload***

  Uploaded files embody a major risk to applications. The foremost step is to get some code to the system to be attacked. Then the attacker has to find a way to get the code executed. We allow only specific file uploads such as jpg, Jpeg, jpe, gif and png.

- ***Email Flooding***

  It is the process of sending large volumes of email to an address to overflow the mailbox, often with large attachments, to disable a network or part of a network such as a mail server. We avoid this kind of attack by removing mailto link in all email IDs.

- ***No-Cookie Attributes***

  Web Cookies are frequent key attack vectors for malicious users, targeting other users and the application should always be attentive to protect cookies. We set Cookie Attributes (HttpOnly and Secure) for the cookies that we use in our WordPress site. This is required to edit the plugin code as well.

Conclusion

Website security is not something to take lightly. Vulnerabilities do not stop and hackers are always finding new ways to hack websites. But, if we implement known security patches for the current vulnerabilities, the chances of getting hacked would be greatly reduced. Hence it is always advised to take regular backups, so that recovery can be successfully made in case of any hacking issues.

Taking security initiatives initially could prevent a lot of damage in the future.

References

https://owasp.org/

https://sucuri.net/reports/19-sucuri-2018-hacked-report.pdf

https://www.netsparker.com/cms-vulnerability-scanner/wordpress-vulnerability-scanner/

https://techjury.net/blog/percentage-of-wordpress-websites/#gref

https://blog.rapid7.com/2017/06/15/about-user-enumeration/#:~:text=User%20enumeration%20is%20when%20a,system%20that%20requires%20user%20authentication

Contact for further details

Selvakumar V

Associate Manager – Digital UX/UI

selvakumarv.in@mouritech.com

MOURI Tech