Wordpress Security Details

FIRST-CLASS SECURITY PROTOCOLS

1.0 WordPress CMS

WordPress is a robust and open source platform that is actively supported by means of continual updates to ensure the best of class integrations, standards and security. WordPress has a dedicated security team which delivers dozens of updates to the CMS core each year. Their rapid update approach eliminates any potential security risks almost immediately.

1.1 Core CMS Security

1.1.1 Up to Date State

Maintaining security is an on-going process, and constant vigilance is essential. Keeping WordPress updated to the latest version is incredibly important to the overall security of web applications. It is true to any piece of software – if a system isn't kept up-to-date, it will be vulnerable to attacks. The dedicated WordPress security team releases security updates regularly (on average 40 updates per year). As part of our monthly support plan, Evoke Solutions, implements those updates within couple of days of their release (there is a preceding testing phase that must happen first).

1.2 Hardening WordPress

Combining proactive and regular maintenance and choosing the right hosting platform, you are able to harden WordPress' core system to be as secure as any other CMS available today.

The best security protocols leverage a layered approach, combining tools and processes that cover all three elements of website security: protection, detection, and response. The following are security best practices which Evoke Solutions brings into implementation when hardening the security aspects of a WordPress web application:

1.2.1 Plugins Up to Date State

In order to empower security, a continuous process to keep all CMS plugins up-to-date is necessary. Each update tends to increase security by patching vulnerabilities and strengthening against attacks and reducing the time frame that the web application is vulnerable.

1.2.2 Forced Strong Passwords & Controlled Access

1.2.2.1 Strong Passwords

The goal with strong passwords is to make it hard for other people to guess and hard for a brute force attack to succeed. The key to this is making complex, long, and unique passwords. As part of our best practice, we force the CMS to accept only strong passwords by default and enforce our client CMS users to update their passwords every 90 days.

1.2.2.2 Password Corporate Policy

Optionally, If there is already a corporate password policy in place, WordPress can be aligned and adjusted according to password policy. Same rules apply across multiple business applications. I.e. Evoke Solutions can build a password validation process based on this corporate policy.

1.2.2.2 User Roles - Limited Access

The least privileged principle builds on the idea that it is about giving people the access they require, for as long as they require to do their job – no more and no less. When the users are done with their work, we reset their access to the most appropriate level. This is most applicable when thinking about users and their appropriate roles. WordPress provides a number of different roles out-of-the-box, each designed with different permission set, but we can customize this as needed.

1.2.3 Secured HTTPS Protocol

The primary reason why SSL is used is to keep sensitive information sent across Internet encrypted, is so that only the intended recipient can understand it. This is important because the information sent on the Internet is passed from computer to computer to get to the destination server. Any computer in between the user and the server can see confidential information like usernames and passwords, and other sensitive information if it is not encrypted with an SSL certificate. When an SSL certificate is used, the information becomes unreadable to everyone except for the server you are sending the information to. Using the HTTPs version of the site is a best of class recommendation of Evoke Solution and we assist in its installation which is very easy with our prefered hosting partner.

1.2.4 Two-Factor Authentication

As an extended and on-demand security, we implement optional “Two-Factor Authentication” for WordPress web application. The idea of multi factor authentication is built on the concept that there is no single solution capable of addressing all security concerns. With a two factor authentication, it is hard to get into the application even in case of a password theft.

This type of authentication requires an additional text along with your password which is basically generated by an external device like your personal mobile installed with app like Google Authenticator on Android or iOS.

1.2.4.1 Web Application Lockdown and Ban Users

Lockdown feature for failed login attempts can solve a huge problem, i.e. no more continuous brute force attempts. Whenever there is a hacking attempt with repetitive wrong passwords, the attacker gets looked out of the site (temporary or permanent as defined in web application) for the specific access point, and administrator gets notified of unauthorized activity.

1.2.4.2 Use Email as Login Identifier Instead Usernames

By default, users have to input their usernames to log in. Using an email ID instead of a username is a more secure approach. Usernames are easy to predict while email IDs are comparatively harder. This feature could be enabled on demand.

1.2.5 Advanced Configuration File Protection

A step ahead in hardening WordPress web application, which is available on-demand, is to separate credential information from the codebase and secure it under more restricted access in same web space or a remote location. Just in case if anyone gets un-authorized access to the main configuration files of WordPress, it does not provide access to more sensitive components of application like a database and server.

1.2.6 Disabling the Default WordPress Online Edit Feature

Disabling the updating of the core application files right from the dashboard which many modern CMS's do provide for the ease of updating the web application content by an administrator. This can help in stopping further damage in case of any unauthorized access to the web application's backend.

1.2.7 Unique Administration/Login Urls

It is very common these days to identify which platform an application is running on and what are default administration/login urls. It then becomes easier for attackers to start targeting such resources of web application. As a part of the ES WordPress Theme Security module we implement following:

- Unique Administration URL per project
- Unique Login URL per project
- Cleaning of the server response headers
- Disable access to the xmlrpc.php file
- Disable the pingback features

1.2.8 Evoke Solutions In-House Theme & Trusted Resources

Being open source in nature, WordPress has a huge library of plugins and theme available on internet to download and start using under free license. It makes WordPress feature rich and enables a rapid development. Since the WordPress core is as secure as any other CMS, it is only through these plugins and themes that hackers gain access to private content of web application in more than 90% of cases.

- Evoke Solutions has decided to build an in-house WordPress theme to have the code under control and leverage the best in class approaches in agile way. The theme is not dependant on any 3rd party provider, it is fully built by Evoke Solutions.
- Use of 3rd party plugins are limited to a minimum and we select only those which are thoroughly tested and have an active support system in place.

2.0 WordPress Hosting

Evoke Solutions recommend hosting provider is WP Engine (http://wpengine.com/), information mentioned below are relevant for this hosting provider.

2.1 Shared Security Features

2.1.1 Disk Write Protection

Malicious code can embed itself into a website by writing to the file-system. This occurs when a vulnerability is present in a theme or plugin that leaves the door open for malicious injection. The hosting environment limits the processes that can write to disk. So even if a theme or a plugin with a vulnerability is used, it is harder for them to be exploited.

2.1.2 Disk Write Limitation

All attempts to write to the disk are logged so that both malicious and non-malicious code are identified.

2.1.3 Disallowed Plugins

Some plugins may expose a website to vulnerabilities. Most of the time, this is unintentional, but still a line in the sand has to be drawn. System scanner searches for these plugins and automatically disables them. Besides disabling plugins for security reasons, plugins can also be disallowed for performance reasons. A comprehensive list of disallowed plugins is being actively updated.

2.1.4 Daily Backups

The whole website; Could be restored anytime via the hosting UI.

2.1.5 Manual Backups

The whole website could be backuped manually anytime via the hosting UI.

2.1.6 Firewall

Firewalls are configured based on the principle of least privilege, where firewalls only allow approved applications, protocols, and services required to meet business needs. Firewalls allow administrator to proactively mitigate external attacks that try to:

- abuse software vulnerabilities,
- brute force attacks that try to break into application backend,
- or denial of service attacks that try to kill the availability of web application.

2.1.7 Network Monitoring

Intrusion Detection or Intrusion Prevention Systems are used to monitor and/or protect the network.

2.1.8 Service availability

Receive immediate notifications through preferred communication channels in case of web application non-availability. It helps to take immediate and necessary steps to bring it back.

2.1.9 Moving from application default settings

As a security best practice, default WordPress configuration(s) during installation are force updated e.g.

- Default database prefix(s)
- Default administrator username
- Disabling server directory listing
- Default robot access to files such as xmlrpc.php and wp-login/admin.php
- Default assets locations/paths to be masked via URL rewrite
- Default nonces/salts in main configuration file

2.1.10 Whitelist your IPs to backend

The application backend can be restricted in such a way that it opens up and listens to only specific IP’s which are internal to business. This would further bring down the chances of brute-force attacks and killing resources since backend is only available to the list of authenticated access points.

2.1.11 Reset File Permissions

After each release of a web application, reset file permissions are issued to ensure all users have proper and relevant access to the files.

2.1.12 Secure File Transfer

Users are allowed to connect to web application’s file system using SFTP only and using very strong password. SFTP uses a layer of encryption for security during file transfer between local machine and remote server.

2.1.13 One-Click Staging Environment

One click staging environment is available in the hosting administration. The staging site runs on a separate domain from the production server and is protected using HTTP authentication - an additional layer of authentication is forced before access to CMS authentication. It is possible to build exactly the same state of application on staging as live to carry out testing activities for each release cycle.

2.1.14 Security Updates and News

WPEngine thoroughly tests new releases and security patches of WordPress as they are available and then recommends to update.

2.2 Dedicated Hosting Plan

To achieve an even more secure environment, a dedicated hosting environment could be selected instead of the shared one. The dedicated hosting environment is a segregated environment (physically and logically) so that data is isolated and protected against any unauthorized access.

3.0 PHP, MySQL, Apache, Ngnix

WordPress as a platform is being written in PHP languages, connected to MySQL database and running in an Apache or Ngnix web servers. Below is an analysis of the security features of these technologies.

3.1 PHP

PHP (hypertext pre-processor) is an open source general-purpose scripting language mainly adapted for web development. It has come a long way since creation and constantly maintained and updated for security and usability. It currently provides a vast array of security features such as;

3.1.1 Allows Retrieval of Environment Variables

With access to capturing environment and server variables, PHP allows you to implement error checking and security functionality with ease and efficiency.

3.1.2 **Built-In Password Hashing Mechanism**

PHP has a built in password salt + hashing mechanism in place which allows you to easily encrypt all user passwords without having to write your own encryption methods by using password_hash().

3.1.3 PHP Code is Hidden from the Public

All the user is able to see when visiting your PHP file through a web browser, is the html that is generated by the PHP. The user will never be able to see the PHP and its functions/methods/variables that are stored on the server in the back-end.

3.1.4 Built-in PHP filters

PHP has a vast range of built in filters to check the validity of objects that you are passing into them. For example you can check that an email address, url, or a number, is valid and is what you were expecting. Not having to write your own escape functions to check these variables adds an extra layer of security and reduces the workload on the application.

3.1.5 Built-in Error Handling and Reporting

PHP has built in error handling and reporting in its core on the fly. Since it is not a compiled language, it allows developers to notice functional and runtime errors before projects go live with ease.

3.1.6 PHP Drivers for MySQL Security

PHP pairs really well with the open source relational database management system (RDBMS) MySQL, as it has many functions to help prevent SQL injection attacks, manage disc I/O and resources on the fly.

3.1.7 Error Logs

PHP stores error logs on the server if errors are to occur, so you can always see where errors/bugs are coming from before the users and 3rd parties do.

3.1.8 3rd Party Libraries

There are countless amounts of 3rd party libraries for PHP that help filter certain attacks such as SQL injection and XSS attacks.

3.2 MySQL

MySQL is an open-source relational database management system (RDBMS) that allows for the quick storage, transfer, and manipulation of massive amounts of data by using the structured query language (SQL). MySQL is constantly updated by the Oracle team to make it more and more secure by the version. With data being one of your most valuable assets, having a secure, accurate, and efficient database management system is crucial.

3.2.1 Users & Privileges

MySQL has a built in privilege system, where to access the database, a user must be first granted access to read, write, or manipulate tables, databases, and data within it. Users can be granted different levels of privileges based on their roles, making it easy to manage security.

3.2.2 Connection Encryption

MySQL has a built in **SSL** library which encrypts the connection and the transfer of data from point A -> B. Making it impossible for hackers to sniff unencrypted data being passed through a network.

3.2.3 Certificate Based Authentication

MySQL has a built certificate based authentication system, which utilizes user privileges and the SSL library, to allow two factor authentication similar to private/public keys on SSH clients.

3.2.4 Database Logs

MySQL has **two** built in logging systems, general log and error log. This allows for DBA's to monitor database security, efficiency, and accuracy without having to do too much searching.

3.2.5 Built-in Bruteforce/Intrusion Detection

MySQL has a built in feature which bans host IP's if too many false authentications are made. This will block future attempts by the same host to try and access your database.

3.2.6 IP Whitelisting

MySQL has a built in feature which allows you to specify and allow only hosts with certain IP addresses or within a specific range in the configuration files (e.g. your web application and nobody else).

3.2.7 Cryptographic functions

MySQL offers many cryptographic functions for retrieving and encrypting/decrypting data on the fly directly within queries if need be.

3.3 Apache

Apache has been the most popular open source web server since its inception. Apache web server is reported to be the most commonly used web server around the world with about 55% of web applications. This number continues to grow every year and with active upgrades and continually new versions and offers secure technology.

3.3.1 URL rewriting

One of the most interesting features of Apache, is the ability to rewrite URLs based on regex patterns to achieve certain perceived navigation. One example, which can be extremely useful when securing your application, is the ability to rewrite specific URLs to use HTTPS based on their relative path.

3.3.2 Using ModSecurity

Apache module, mod_security works as a web application firewall and empowers web security to application. Different functionalities include filtering, server identity masking, and null byte attack prevention. Real-time traffic monitoring is also allowed through this module.

3.3.3 Allow Secure Protocols

Apache has built-in support through mod_ssl module to add an SSL certificate to your site which is the most widely known protocol that offers privacy and good reliability for client-server communication over the Internet.

3.3.4 Monitoring Through Log Files

Apache provides a way to log all incoming application traffic. It helps analyze of past events and gives the administrator a good idea of what attack trends are being followed so the application security can be tightened up accordingly.

3.3.5 Wide Range of Native Directives

Apache has wide range of in-built modules and directives to restrict traffic based on request size & IP address, limiting concurrency, lower the Timeout, adjust KeepAlive settings, http basic authentication - which gives web administrators an upper hand to set security rules.

3.3.6 Virtualhosts

Another helpful feature of Apache is that it allows administrators to create virtual hosts so the application can be logically broken down into smaller entities and hosted separately from each other.

3.4 Nginx

Nginx HTTP Server is an event-driven, asynchronous, nonblocking, and free open source server which is highly recommended and the second most popular open source HTTP server due to its high-performance, sustainability, premium features, easy configuration, and unmatched speed.

3.4.1 Proxy Server Capabilities

Nginx can be used as front-end proxy servers and then apply a set of rules to incoming request before forwarding further to back-end servers. e.g. permanent redirects to https from http. It also helps improving security as backend servers can be made accessible via a single public IP address

3.4.2 SSL Support

Nginx has in-built module that provides the necessary support for HTTPS.

3.4.3 Load Balancer

Nginx has native support and secure techniques for optimizing resource utilization, maximizing throughput, reducing latency, and ensuring fault-tolerant configurations.

3.4.4 Mod Security Support

Add additional layer of security by implementing Web Application Firewall ModSecurity.

3.4.5 Wide Range of Native Directives

A range of options to further hardening security aspects e.g. disable unwanted methods, XSS Protection, IP restrictions, web traffic monitoring and control mechanism etc.

3.4.6 Different Packages and Active Professional Support

Nginx comes in different packages based on the web application and security requirements like nginx-core, nginx-light, nginx-full, nginx-extras, nginx-naxsi, and the ability to remove unwanted modules. For enterprise level options, Nginx comes with active professional support.