Routine Website Technical Maintenance

Information For My Clients

Bob Spies, Flying Seal Systems

Updated: 25-Apr-2020

This document has several purposes:

- To explain what routine website technical maintenance is and why it's critical that you do it.
- To describe the routine website technical maintenance service I provide to clients.
- To explain why I recommend that most clients use my maintenance service rather than attempt to do their own maintenance.
- To give some pointers to clients who choose to do their own website maintenance.

Introduction

Routine website technical maintenance for a WordPress site consists of **backups**, **routine software updates**, and **security protection and monitoring**. I offer a website maintenance service to clients that covers these activities for a relatively small cost.

All three of these activities are often short-changed by website owners until something goes wrong. At that point, the results can range from inconvenient (something stops working or your website goes down) to catastrophic (an intruder modifies your website to steal money from your clients).

Routine website technical maintenance is no different than auto maintenance: you need to perform it regularly to avoid problems.

The sections below include some pointers that will hopefully be helpful to those who choose to do their own maintenance. However, they are not intended nor are they sufficient to be a complete reference or how-to guide.

Backups

Most website hosting companies do not provide adequate backup for your site. The host's backup capability--if it even exists--only allows you to restore the entire account to the state it existed in at the time of the backup. Getting the host to do the restore can be problematic. Finally, the quality/integrity of a host backup and/or restore may be an issue.

What I Do For My Maintenance Clients

For my maintenance clients, I run backups at least daily, and more frequently for clients who need that. I keep a week worth of rotating backups on the site, and once a week download the most recent backup for offline storage. I test the integrity of all downloaded backup archive files. I keep all offline copies for at least 90 days, and a subset of them for much longer than that.

If You're Choosing to Do Your Own Backups

You'll need to install backup software. There are free alternatives available, although I don't consider them as safe/reliable as UpdraftPlus (paid version), the software I use for my maintenance service.

Used properly, a backup plugin can be a helpful part of your backup strategy. However, it is only a tool. It requires proper configuration and monitoring. **The fact that it's installed on your site does not by itself mean that your site is safely being backed up.**

When using a backup program:

- Make sure you understand and correctly set the options, including setting up an appropriate schedule.
- Make sure your backup strategy includes copies stored in a safe place (not on the web server).
- Make sure you're monitoring that backups are happening when expected. Periodically test the integrity of the backup archive files. (It's best to test the offsite copies, since the transfer process is one place where corruption may occur.)

Why It May Be Better To Let Me Do It

The steps involved in properly setting up backups, making sure they're running when they're supposed to, and validating the backup files are time-consuming and require technical expertise. For example, the most reliable way to copy the backup files from your website to somewhere else is SFTP, which isn't particularly user-friendly. When a restore is necessary, analyzing what's needed and the best way to accomplish it can require a fair amount of technical expertise. Performing the necessary monitoring to ensure scheduled backups are happening properly is something that's easy to forget to do.

I have the necessary technical expertise to deal with the issues that may come up. And since I'm doing this for multiple clients as a business activity, I have a schedule, systems, and procedures in place that make it more likely that all the necessary steps will happen correctly and on time.

Routine Software Updates

On a WordPress install, the following components get routinely updated:

- WordPress Core
  - Major Updates – These have version numbers like 5.2, 5.3.

    - Minor Updates – These have version numbers like 5.21, 5.22, etc., and are generally security related. Unlike most other updates, these (a) happen automatically, and (b) are very unlikely to break anything on your site.

- Theme (controls the site's visual aspects). With most sites I build, there are likely to be a parent theme—most commonly Genesis (technically a theme framework)—and a child theme, which is where I've done most of the custom work. The parent theme is the one that periodically requires updates. (Updates are occasionally required for child themes too, but this is rarer.)

- Plugins (3rd party code that adds additional functionality to the site).

There are three reasons for these updates:

1) New features and functionality.
2) Continued compatibility with other software that has been updated (e.g., WordPress core).
3) Resolution of newly discovered security vulnerabilities.

Sometimes website owners believe they can forgo updates because they don't need new features and functionality. But that ignores reasons (2) and (3) above. The bottom line is that you need to keep up with updates or risk (a) your website malfunctioning, or (b) even more important, your website becoming infected due to a security vulnerability. With respect to the latter, shortly after a security vulnerability becomes public, malware engines around the world get updated. These systematically and randomly probe websites looking for the vulnerability. They then infect vulnerable sites.

But even though it's important to keep your website up-to-date, occasionally an update itself will cause problems. It's not possible for an update's author to be able to anticipate all the situations that exist on every website on which it's going to be applied.

So, you need to:

a) Always be ready to revert the application of an update. There isn't any "standard" way of doing this; it's a matter of restoring relevant components from a recent-enough backup.
b) When it appears that an update is likely to be "high-risk" on your site, test it in a test environment first.

Most updates can be applied from your WordPress admin using a simple "one click" process. What's not so simple is dealing with any problems the update causes.

What I Do For My Maintenance Clients

I use my knowledge of the updates waiting to be applied to determine the level of risk associated with each.

I make sure there is a backup available in case an update causes problems.

In the rare event that an update process fails, I clean up the damage and apply the update manually.

If I judge there is enough risk to merit it, I first test the update in a test environment.

If an update causes problems, I fix them and/or revert to backup.

- If necessary, I analyze the source of the problems and make changes to the client's setup so the update can be applied. If it appears that the update itself is defective, I work with the vendor if necessary and wait for a fixed version. If applying the update requires extensive changes to a client's site, there is the possibility there might be chargeable time involved. I would always clear this with the client first. To date this has been an extremely rare occurrence.

If You're Choosing to Do Your Own Routine Updates

Make sure you have a current backup before performing the update.

Install one update at a time. That way, if something goes wrong, you'll know which update was responsible.

Unless the notes associated with an update say it fixes a critical security vulnerability, consider waiting a week after it comes out before installing it. That gives time for an update-to-the-update to be issued if any significant problems with it are found.

Consider learning how to create and use a test environment. Most hosting plans allow this for no additional cost. However, it can be complex technically.

Why It May Be Better To Let Me Do It

If something goes wrong, I'm in a position to deal with it immediately so your site isn't down or compromised.

Also, I'm generally familiar with what's going on with current widely-used updates. I'm therefore probably going to know how much likelihood there is of a particular update creating problems.

Security Protection and Monitoring

The web has become a dangerous place, both in terms of the number and sophistication of attacks on websites. Whereas a few years back the concern was primarily lone hackers looking to do malicious mischief, now the major concern is automated attack engines with an objective of theft. Today's attack may consist of a hidden installation of code on your site designed to infect your visitor's PC's with ransomware or phish their credit card information.

What You Need To Do Regardless of Who Does Your Maintenance

Don't create a site admin user id that contains either the word "admin" or the name of your website.

Use secure passwords. WordPress can now auto-generate passwords that are extremely secure.

Don't access your site admin over public wifi unless your site uses https or you're using a VPN.

What I Do For My Maintenance Clients

I install the Wordfence security plugin, and use it in several ways beyond its default protection:

- I configure it to protect against "brute force" attacks on your site.
- Wordfence notifies me of various conditions identified as a result of its daily scan. This includes differences from repository versions in core WordPress, themes, and plugins. I review any alerts and determine whether there's a problem.
- Wordfence notifies me of all logins on your site by users with administrator privileges. I review the user name and location they're logging in from to look for anomalies. (This helped me to catch a dangerous break-in on one client's site almost immediately.)

Why It May Be Better To Let Me Do It

I have Wordfence configured to identify various things that might need attention. An example is flagging when a theme or plugin file differs from the theoretically-same version in the WordPress repository. I can look at the differences and determine whether or not they represent an issue. Usually they don't; however, over the time I've been running my maintenance service there have been two occasions where they were associated with an infection.

I hope this document has been useful. If you have any questions about it, please feel free to contact me here.