Hardening WordPress

Essential guidance for anyone running a self-hosted WordPress website

Andrew Villeneuve
Cybersecurity Strategist

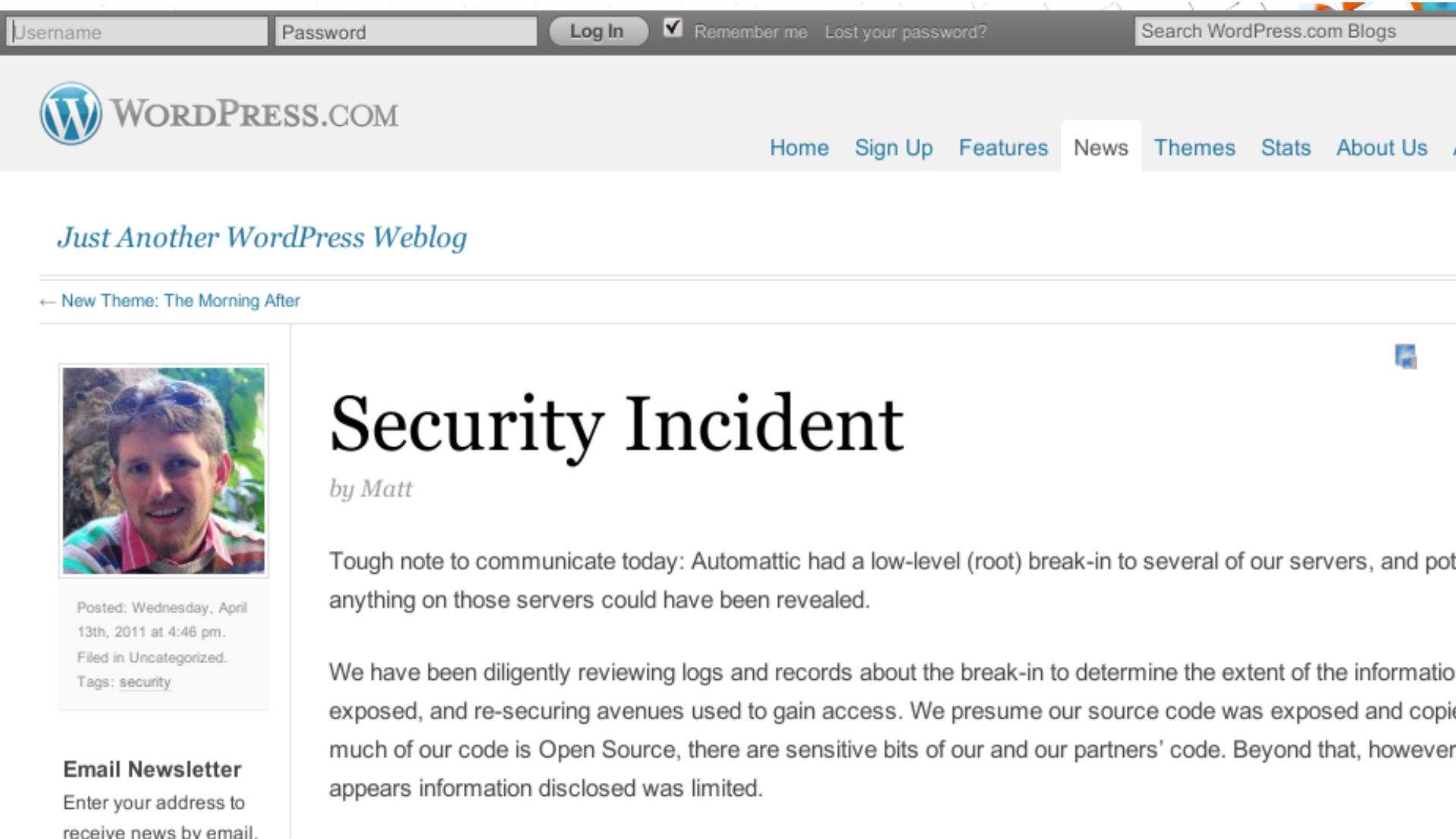
Credits and Copyright

About the author

Andrew Villeneuve is an independent cybersecurity strategist who has been securing WordPress websites, rescuing compromised computers, and helping people stay safe online for nearly a decade. Andrew has led many security trainings and talks about cybersecurity and WordPress security. He resides in Redmond, Washington.

Contact Andrew at www.andrewvilleneuve.com.

Staying secure is important!



Step 1: Choose a good host

- The security of your site will depend in part on the strength of the server environment it is in. You don't want to be on an outdated server stack that's vulnerable to problems like Heartbleed, Shellshock, or POODLE
- When it comes to hosting, you have choices. Managed hosting is the traditional kind. Unmanaged hosting is becoming popular.

Going the unmanaged route

- Some hosts are bad at keeping their stack updated, so going the unmanaged route can be good from a security perspective. **But you have to remember to update**.
- Popular choices for unmanaged hosting these days include Amazon, Linode, UpCloud, and DigitalOcean. If you want to simplify your unmanaged hosting experience, there's services like ServerPilot or RunCloud that can help automate server management.

Don’t choose unmanaged hosting unless you have a long-term relationship with a developer

- If you don’t know what the terms *command line*, *virtual private server*, and *LAMP stack* mean, then you should stay away from developer-oriented hosts like DigitalOcean, AWS, UpCloud, and Linode… unless you have a relationship with a developer who will maintain your site(s) for you.

Sticking with managed hosting

- Be picky when it comes to hosting! The reliability and value of the service are obviously important considerations – but what else should you look for?
- **How modern is the server stack?** Look for a host that supports the latest versions of PHP, Apache, and MySQL.
- **How transparent is the host?** Do they have a status blog where they report their downtime for the world to see and announce their maintenance plans?
- **Is SSH readily available?** The command line is a developer's best friend. You shouldn't need to fax in a copy of your driver's license to get SSH access.

Specific security-related questions to ask a host before you sign up with them

- **Do you offer SSH & SFTP access?** (Answer should be yes).
- **Do you offer secure hosting without requiring purchase of a unique IP?** (Answer should be yes.)
- **Do you allow your customers to install self-signed certificates and professionally-signed certificates purchased elsewhere?** (Answer should be yes.)
- **What version of PHP are your servers running?** (The answer, circa 2019, should be at least PHP 7.1.x or 7.2.x.)
- **What version of MySQL are your servers running?** (The answer, circa 2019, should be 5.6.x or higher.)

Sticking with managed hosting

- Lots of options to choose from in this category. SiteGround has become popular. I like DreamHost.
- WPEngine has become a popular choice for folks who want most things taken care of for them.
- Many hosts are owned by two large conglomerates: GoDaddy, Inc. and the Endurance International Group (EIG). If you prefer to do business with a customer centric host that isn't just a subsidiary of a huge company, then steer clear of MediaTemple, HostGator, BlueHost, etc.

Backing up and upgrading

Step 2: Set up automatic backups

- Back up regularly.
- Learn how to back up your WordPress site yourself manually if you don’t know how.
- Don't rely on your host's backup solution.
- Once you know how to back up, set up automatic backups that you control. Don’t rely on a manual backup strategy. It doesn’t work and it’s too time consuming (especially when you have a lot of sites) .
- Invest in an off-site backup solution. **Off-site backup is your insurance**.

Backup solutions

For local and offsite backups:

- Updraft Plus: This is what I use on my sites and recommend to clients.

Alternative for offsite backup:

- VaultPress: Part of Jetpack, offered by Automattic (the folks who develop WordPress and run WordPress.com)

Need to backup manually before an upgrade? It only takes one command…

```
user@remote:~$ cd sites
user@remote:~$ tar -zcvf may-19-2012-backup.tar.gz mywpsite.com
user@remote:~$
```

… to backup the filesystem.

If you don't see an error message, a tar.gz archive **was successfully created**. You can type "ls -a" to see it.

```
user@remote:~$ ls -a
user@remote:~$ mywpsite.com  othersite.com
thirdsite.com  XXX-XX-XXX-backup.tar.gz
```

Step 3: Update, update, update! Different ways to update

- Download from WordPress.org and then manually upload new filesystem (ugh)
- Use your web host's one-click upgrader (if there is one available)
- Use the built-in updater that comes with WordPress nowadays (automatic updates are enabled by default for minor releases)
- **Or use WP-CLI!** `user@remote:~$ wp core update`

Don’t put off upgrading!

- There is no good excuse for not upgrading. Don't wait. Don't leave your sites and your client’s sites vulnerable to known exploits.
- As WordPress founder Matt Mullenweg says... *A stitch in time saves nine!*
- Keep plugins upgraded.
- Keep themes upgraded.
- Ask your developer to put any customizations to a stock theme in a **child theme**.

Step 4: Utilize HTTPS

What is HTTPS? What can it do?

- HTTPS = HTTP Secure (Hypertext Transfer Protocol with Transport Layer Security)
- What it can do: Stop man-in-the-middle attacks, thwart eavesdropping
- WordPress has two built-in HTTPS modes for securing the backend (wp-admin). One forces HTTPS for logins. The second mode forces HTTPS for administrative sessions.
- Use the second mode! There's no good reason not to. It offers better protection.

Enabling HTTPS

- Either of the WordPress HTTPS modes can be turned on by adding just one line to the wp-config.php file. No plugin required!
- But there are prerequisites.
- **Set up secure hosting with your host. You will need a secure certificate**. They're free from Let's Encrypt. Secure certificates that are compatible with many older crumbly browsers are also available for as little as $8/year.

Turning on HTTPS in wp-config.php

Only one line of code needs to be added:

```
define('FORCE_SSL_ADMIN', true);
```

While you're editing your wp-config.php file, make sure that there are secret keys in place! The keys should look like this:

```
define('AUTH_KEY',         '.LxcZk#k|m+:-y:!+zz$w7dIW<|.WB+A%jx|.8!AhmvA7!$$J=TA9<_mk@:RAwH2');
define('SECURE_AUTH_KEY',  '&m|,l#C|+dC-O+Qi3%|U?HT%[& -i<mAyd2a}i_D=oj~LX<${)JN87{s9ia5rQek');
define('LOGGED_IN_KEY',    '%4Dt-7=V`s<.(G+B|PbO4=7-!8Orm}<?|W-a{X43g(r(/~_8KJ!oUJ$aT!5CY-_j');
define('NONCE_KEY',        '#X#`nUS$:4-MW2!9W^-EFr>ww<:eE^6DIT0d)%:GBS388I7.+-hdEUktC}v8jEpM');
define('AUTH_SALT',        'E fx:ie$7kfhJ}sw|S^C:a2Y4!vkv(0W$p-rch:vr`|x |f2&|!zP|r2|;k7(?]f');
define('SECURE_AUTH_SALT', '0EGtyq5*|Sy66-:+7)R4c73.1y)j8Q<gLcF8Zt{~d?n7#)FrltPZ/p/wcJU~r(iN');
define('LOGGED_IN_SALT',   'dIRc|ww+U{+#;H^Vi3b`+L7(B6#le)-.56B>{7h/m}6{7-0aVC07<D{8:gvw)FMD');
define('NONCE_SALT',       '|+6iVllc5~x?_Oiml)O+u~+r:7cJ+u;U>+}|3S?DP[Eg5P/<3 d~QFSz[)+{1S%+');
```

Use HTTPS on your frontend, too!

- You can force HTTPS on your sites' frontends, too. But first you need to set up secure hosting, update URLs in the site's WordPress database to have an HTTPS prefix, and eliminate mixed-content warnings by updating references.
- When you are ready, set the site and home URLS like so in the WordPress Settings:
  - https://www.mysite.com/
  - https://www.mysite.com/
- Then, instruct Apache, nginx, or whatever server you're using to redirect HTTP traffic to HTTPS. Recipes to do this are readily obtainable.

Step 5: Tighten permissions

- Keep key files in your filesystem locked down. Otherwise, it's easier for hackers to mess around if they manage to break in.
- Ideal permissions:
  - root folder: 705
  - wp-config.php, index.php: 400
  - .htaccess: 404 (set **after** configuring firewall)
  - wp-includes/, wp-admin/, wp-content/: 705
  - wp-blog-header.php: 400

Step 6: Adopt good password hygiene

- If you are making up a password yourself, use at least half a dozen letters. **Mixed-case** is good. Use random letters or uncommon acronyms only. **Do not use words**. *If it's in a dictionary... DON'T USE IT!!!*
- **Numbers**. At least a few integers (0, 1, 2, 3, 4, 5, 6, 7, 8, 9). More is better.
- **Punctuation**. Punctuation is essential in a strong password. WordPress will let you use pretty much any punctuation mark that you see on your keyboard.

Example passwords

- **Weak**: `mydogbailey12`
- **Strong**: `hfMdby12/aowh;03`
- The second password is a variant of the first with more complexity. hf = happy family, Mdby = my dog bailey, 12 years old, aowh = adopted in old wonderful house, in the year 2003.
- The key? Use uncommon acronyms that would be meaningless to other people, but that are memorable to you.

Really, *really* bad passwords

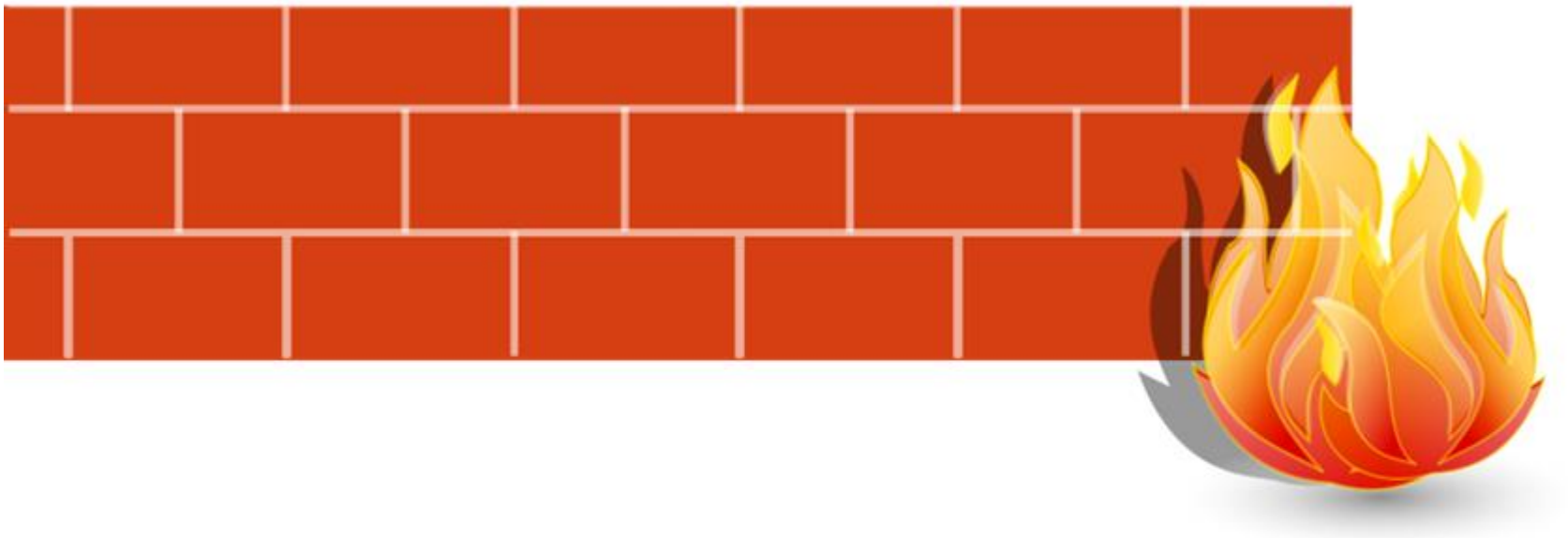
- P@ssw0rd, Password1!
- 10Topsecurity-
- BusinessName12
- may191968
- 4257770707
- admin

Again, don’t use words. Don’t use words where some of the characters are replaced by punctuation, either. That’s not a good practice!

Please use a password manager!

- It is better to let a password manager generate your account and application passwords for you. (I recommend Dashlane.) Set a strong master password using the tips provided on the previous slides. Protect your manager with multi-factor authentication (also known as 2FA).
- A manager can help you avoid recycling passwords. Never, ever use your MySQL user password for other things, and never use your WordPress password for other things.

Step 7: Building a firewall



Setting up a firewall for WordPress

- Building a firewall against attacks is of paramount importance. The thing is, manually configuring .htaccess on **Apache webservers** can be time-consuming and difficult.
- That's where **BulletProof Security** comes in. It's a plugin that'll do the cumbersome work for you.
- BulletProof has a setup wizard that will automatically configure a working application firewall at the webserver level.

Step 8: Stop spam and by…

- Installing Akismet
- Automatically closing old comment threads
- Keeping robots.txt rules up to date
- Password protect directories that have files with private/confidential information. Anything that you wish to restrict address to should at least be password protected. **Remember, security through obscurity doesn’t work.**

Accessing your WordPress

- As a general rule of thumb, try not to access your WordPress admin from shared or public computers. You don't know what's on them.
- Do not log into your WordPress admin at a public wi-fi hotspot unless you are using Forced HTTPS mode to encrypt your session.
- Don't borrow somebody else's computer to access your WordPress installation unless you trust that person and can verify that the computer is malware-free.

Accessing your WordPress, Part II

- Keep your computer clean. Make sure an internet security suite is installed (firewall + antivirus). According to independent labs, the top suites for Windows and Apple's macOS are made by Kaspersky and Bitdefender.
- Put **armor on your browser**. Do not enable JavaScript or cookies by default. Only enable scripting for sites that you trust.
- WordPress has apps available for both major mobile platforms. Take advantage!

Lastly: Test your configuration

- **Scan for malware externally**: You can use the Sucuri Malware Scanner.
- **Test the security** of your server environment: You can use Hardenize.
- **Perform a deep analysis of your server's** secure hosting, including your certificate: You can use the service offered by Qualsys Labs.