Higher Order WP Security

Hacks, attacks, and getting your site back

Dougal Campbell

I totally stole that image from another site. Sorry.

HACKERS!

Everybody know what a ‘hacker’ is, right? The bad guys. The people who want to break your site, or sneak nasty links into it, or use your site to infect others with viruses.

HACKERS!

CRACKERS!

But most programmers, especially old-school ones like myself, call themselves “hackers”. It’s sort of a Light Side vs Dark Side thing. Or “white hat” vs “black hat”, as you’ll sometimes see. There was an effort to differentiate the more generic term “hackers” from the more insidious “crackers”, as a term more specific to the bad kind of hacking.

HACKERS!

Everybody says “hackers” anyways.

Buuuut... Everybody still says “hackers” anyways. Think of it like the slang words “bad” or “wicked”, which were repurposed to mean the same thing as “cool”. Which often meant the same thing as “hot”. Language is fluid, roll with it. During this presentation, I’ll *mostly* be using “hackers” as a synonym for “crackers”.

WordPress Hacks

- Warning! Massive Number of GoDaddy WordPress Blogs Hacked!
- DreamHost: One Million Domains Hacked; WordPress Blogs Infected
- WordPress Sites on GoDaddy, Bluehost Hacked
- Reuters Hacked Again, Outdated WordPress Blog At Fault?
- InMotion Hosting Servers Hacked, Thousands of Web Sites Affected

Thursday, September 27, 12

You've probably seen headlines like these over the past several years. Reports of thousands or even millions of web sites hacked, with WordPress mentioned prominently. Boy, WordPress must be an insecure mess, huh?

WordPress Hacks

- History shows there have been very few “WordPress Hacks”
- *“ In the vast majority of cases I see, attackers get in some other way, and then once already in the system, they go looking for WordPress installs.”* -- Mark Jaquith

Truth is, WordPress has hardly ever been the weak point in these hacks. It was a symptom, not a cause. A runny nose doesn’t cause your cold, it’s a symptom. Likewise, a hacked WordPress site does not mean that WordPress caused a site to get hacked. The “infection” often starts somewhere else.

If WordPress isn't the weak point, what is?

WordPress Hacks

- Most hacks that **affect** WordPress actually originate **outside** of WordPress Core.
  - TimThumb (PHP library, many themes/plugins)
  - Uploadify (jQuery plugin, many themes/plugins)
  - Adserve (plugin)
  - WassUp (plugin)
  - Is Human (plugin)

Thursday, September 27, 12

Often, the weak points are third-party plugins and themes for WordPress.

The TimThumb library is a famous recent example. This library is very popular for allowing you to manipulate images (crop, rotate, resize, etc) right in the browser. Many WordPress themes and plugins integrated this library. When it turned out that TimThumb had a security weakness, MANY WordPress sites were affected, and were in fact hacked. WordPress itself was not the problem. Sites which did not have plugins or themes that used TimThumb remained safe, because WP Core was not the weak point.

TimThumb has since been fixed, but it suffered a major black eye due to how many sites wound up being affected.

Likewise, Uploadify is a jQuery plugin which had a security weakness affecting some plugins and themes. Adserve, WassUp, and Is Human are WordPress plugins that all had some sort of security weakness that was discovered and used by hackers to attack sites.

We need to look at the bigger picture

But even if your plugins and themes are safe, or even if you don’t run any extra plugins, that doesn’t automatically make you safe. You need to expand your security awareness beyond WordPress Core, beyond themes, and beyond plugins.

The LAMP Stack

WordPress is an application that sits on top of what we most often call “The LAMP Stack”. LAMP stands for “Linux, Apache, MySQL, and PHP”, a very common web server setup. Some sites might use something other than Linux as their operating system (FreeBSD, SunOS, etc). Or they might use Nginx instead of Apache for the web server. But in any case, there will still be these basic elements: an operating system, a web service, a database, and a programming language, with a web application like WordPress sitting on top of them.

ANY of these pieces could potentially have security flaws in them that could allow an attacker to compromise your web site. If you look up the history of software releases for each of these things, you might be surprised at how often releases contain fixes for security-related bugs. Most of the nasty ones were shaken out a long time ago. We hope. But were talking about very complicated systems with many thousands, or even millions, of lines of code. Sometimes a small oversight can be turned into a crack in the armor by a clever hacker.

Other Services and Apps

- SMTP (email)
- FTP
- DNS
- Other web sites and utilities?
  - Drupal, Joomla, forums
  - PHPMyAdmin

In addition to WordPress, there maybe be other network services running. SMTP for receiving email on the server; FTP for file transfers; most people don't run their own DNS service, but some do; and you or others on the same server as you might run other web apps like Drupal or Joomla, forums like bbPress, phpBB, or Vanilla, or a database management utility like PHPMyAdmin.

Any one of these things on your server could contain some security flaw that an attacker could leverage to gain access to other parts of the server.

Shared Hosting

- Shared hosting? Shared security!
- Other users on the same server as you can become a security risk that affects you
- What about your own users? Can you trust everyone who has a login for your site? *Really* trust them?
  - "Nobody cares as much about the survival of your business as yourself." -- Ron Cain, business owner

Thursday, September 27, 12

Many people have their sites on shared hosting. If you aren't on a "dedicated server" or VPS (virtual private server), then you are almost certainly on shared hosting. That means that there are other users running web sites and other services on the same server as you. They are normally segregated out so that users can't "see" each other (usually through "jailed" or "chroot" environments) that offer some protection, but ultimately, these users are all sharing some resources at the operating system level.

But even if you are using a totally dedicate server, where nobody else has access to the operating system, what about other WordPress users? If you have a multi-author site -- really any situation where other users login to WordPress and see the Dashboard -- you have to know how much you can trust them with your site.

My wife's parents own a small business. Over the years, when they have hired part-time employees to help out, one problem they've had is that some people just didn't follow through with all the things they were supposed to do. You might tell them that during slow times, they are not supposed to use the Point of Sale computers for surfing web sites, but when the owners are away, and they aren't taking care of customers, that's what some employees would do. And while they aren't trying to be problematic, they also aren't the ones who will have to deal with the cleanup if those computers became infected with a virus. Because they are not the ones who invested their savings into the business, they will probably never care as much about the business and its assets as the owners do.

It's the same with your website. Whether you use your site for e-commerce, or as a "brochure" business site, or just for a hobby, it's unlikely that any other user of your site is going to care as much about it as you.

How do hackers get in?

- Known exploits in vulnerable software
- Brute-force password hacking
- Network scanners
  - Firesheep
  - Wifi vulnerabilities (WEP/WPA)
- Automated tools
- Rootkits

Hackers often share their knowledge of security holes with other hackers. In those circles, there are big “bragging rights” for being the first to discover such a hole in a piece of software. In InfoSec circles, there are “white hat” and “black hat” hackers. The white hats discover security holes either as part of their jobs as security consultants, or as a hobby. But when they find something, they typically send a security report to the owners of the software, and give them a window of time to create and release a fix before making the knowledge of the hole public. The black hats, on the other hand, just start using the security hole to break into systems. And at some point, they share information about the holes with others. Once a new exploit becomes known, there is usually a spurt of activity as more and more hackers begin testing the hack against servers.

One common target once a hacker gets into a system is to obtain access to account passwords. In most cases (with a few lamentable exceptions), passwords are encrypted, or “hashed”, to protect the passwords from this type of unauthorized access. You might type in your password as “syncronicity” (don’t use that), but in the password database, it will be encrypted and stored as something like “93fd5f81657c05b5a6e485ae216313e3e092f44c”. To match this up with the original password, a hacker can run a program that will try every combination of letters, numbers, and symbols, encrypting each combination, looking for a match. A word like “syncronicity” is EASY to crack, because it appears in the dictionary. Password crackers keep lists words like that, pre-encrypted with a variety of password hashing algorithms, and can find matches for them EXTREMELY quickly. More on password security later...

Network scanners will monitor network activity, and can nab unencrypted passwords right out of the network packets. Firesheep was an extension created for the Firefox web browser to demonstrate just how easy this could be in some cases. It would monitor the network you were connected to, watching for others to login insecurely (without SSL/https) to services like Facebook, Twitter, and the like. It would then present you with a button in your browser that would let you simply click to login as that other person. It was scary how well it worked. Since

Staying Safe

Now that I've scared the crap out of you about all the ways that bad guys can break into your systems and make your life miserable...

What can we do to stop them?

Three Words

Thursday, September 27, 12

Three Words

- Update

Three Words

- Update
- Update

Three Words

- Update
- Update
- Update

Three Words

- Update Core
- Update Plugins
- Update Themes

Keep your WordPress Core, plugins, and themes up-to-date. When you login to your WordPress administration Dashboard, it will show you when updates are available.

You will practically never break anything on your site by upgrading WordPress Core between minor versions (e.g., from 3.4.1 to 3.4.2). Minor releases are only for bug fixes, and will never contain any major new features. Major releases, say from 3.4 to 3.5, are the ones that contain new features, and possibly new bugs. Some people like to wait a couple of weeks after a new major release before upgrading, just to allow more adventurous users to find the bugs first. :) (not me, I run many of my sites right out of the development trunk, updating daily).

When plugins and themes update, read the release notes and see if you need to upgrade them. In most cases, you will want to. Sometimes you might see something that isn't crucial to you, like "upgraded translations to add the Estonian language". In that case, you might be able to just leave it and wait for some future upgrade. Unless you have a lot of Estonian users, of course.

What Else?

- Hotfix Plugin
- WP Security Scanner
- Login Lockdown
- BulletProof Security
- Sucuri.net

The Hotfix plugin is maintained by some of the top WordPress developers. It attempts to give you bugfixes that might have been found, but not included in an official release yet. It is always safe to have this plugin active on your sites.

WP Security Scanners (and several others similar to it) will examine your files for certain types of known problems, and give you a report of potential problems.

Login Lockdown, and other similar plugins like Limit Logins, will watch for brute-force login attempts, and block a client after too many failed login attempts.

BulletProof Security modifies your .htaccess file to block certain types of access attempts at the web server level (Apache), before WordPress can even try to handle the request.

Sucuri.net offers security monitoring, auditing and cleanup services, primarily for WordPress.

What Else?

- Not using a plugin anymore?
  - Deactivate
  - DELETE!
  - The same goes for themes

Do you have old plugins or themes that you don't use any more? You may have installed several themes, looked at them, finally found the one you liked, and just never got rid of the the others you tried. Or you switched from the Awesome Tweets plugin to the Super Awesome Tweets plugin (I made those up), and the old one is still hanging around, deactivated. Or you have a plugin that's active, but you just never use it. GET RID OF THEM. Even if the plugin is deactivated it could still offer an avenue of attack, if it is written badly.

Imagine if you will: http://mysite.com/wp-content/plugins/some-old-plugin/sop-admin.php

If such a file was written without security in mind, direct access like this could bypass WordPress and what it thinks is active or inactive.

HACKED!

Oh dears.

We were too late. We didn’t secure our site, and we’ve been hacked.

NOW WHAT?

Now What?

- You can no longer trust any code files
- Nuke the site, start from *trusted*, fresh copies
  - Save wp-config.php and wp-content/uploads
- Reinstall data from backups

Thursday, September 27, 12

Every PHP file in your site is now suspect. Hackers could have added files that let them control your server in all sorts of ways. They could have hidden evil code in any one of the hundreds of files in WordPress core, your themes, or plugins. This goes for JavaScript, too.

You need to delete everything except wp-config.php and your wp-content/uploads folder, and reinstall WordPress, themes, and plugins FROM TRUSTED SOURCES. Even your own recent backups should be treated with suspicion. Hopefully you backed up any custom plugins or themes before your site even went public. And if you're saving your wp-config file, double-check it before you put it back up.

Now What?

- You can no longer trust any code files
- Nuke the site, start from *trusted*, fresh copies
  - Save wp-config.php and wp-content/uploads
- Reinstall data from backups
- You *do* have backups, right?

Thursday, September 27, 12

Every PHP file in your site is now suspect. Hackers could have added files that let them control your server in all sorts of ways. They could have hidden evil code in any one of the hundreds of files in WordPress core, your themes, or plugins. This goes for JavaScript, too.

You need to delete everything except wp-config.php and your wp-content/uploads folder, and reinstall WordPress, themes, and plugins FROM TRUSTED SOURCES. Even your own recent backups should be treated with suspicion. Hopefully you backed up any custom plugins or themes before your site even went public. And if you're saving your wp-config file, double-check it before you put it back up.

Now What?

- You can no longer trust any code files
- Nuke the site, start from *trusted*, fresh copies
  - Save wp-config.php and wp-content/uploads
- Reinstall data from backups
- You *do* have backups, right?
- ***Right***?

Thursday, September 27, 12

Every PHP file in your site is now suspect. Hackers could have added files that let them control your server in all sorts of ways. They could have hidden evil code in any one of the hundreds of files in WordPress core, your themes, or plugins. This goes for JavaScript, too.

You need to delete everything except wp-config.php and your wp-content/uploads folder, and reinstall WordPress, themes, and plugins FROM TRUSTED SOURCES. Even your own recent backups should be treated with suspicion. Hopefully you backed up any custom plugins or themes before your site even went public. And if you're saving your wp-config file, double-check it before you put it back up.

What do I back up?

- Database
- Uploaded media (wp-content/uploads)
- Custom themes and plugins
- wp-config.php
- Keep a list of your installed third-party plugins

Thursday, September 27, 12

Your content is your site's lifeblood. Even if you don't back up anything else, back up your database. In a worst-case scenario, you could at least get *something* up as a new site. Even if you don't have the same theme or plugins, you could at least have some sort of basic site back up.

Also, any uploaded media -- images, audio, videos, screencasts, PDF ebooks that you make available -- make sure you have a backup of the originals. And it's going to be easiest if you can just back up the entire wp-content/uploads directory as-is, so that you don't have to re-upload everything from scratch.

Did you create, or hire someone to create, any custom theme or plugin work? Make backups of those as soon as they are created.

You probably still have access to the information about your database configuration and other settings, but it's easiest of all if you just have a good backup of your wp-config.php file.

You don't have to back up all the files for your plugins. But at the very least, save a list of which plugins you have installed. If nothing else, take a screenshot of your Active Plugins page.

NOTE: Some people have asked if saving a copy of a WordPress export file (the .WXR file created by the export tool) is as good as backing up the database. In a word, no. The export file is great for migrating your content, categories, and tags to a new server. But it is not the same as a full database backup. It *only* contains your content. It does not contain other crucial information like the configuration of your site settings and plugins.

How do I back up?

- Backup Buddy
- VaultPress
- WordPress Backup to Dropbox

Thursday, September 27, 12

These are just a few of the options available to help you back your site up.

BackupBuddy will back up your database, themes, plugins, etc. It can store backups in cloud storage like Amazon S3, Rackspace Cloud, and Dropbox, or send the backups to you in email, send them to an FTP server, or let you download them straight to your computer. Commercial, not free, but many people swear by it.
http://ithemes.com/purchase/backupbuddy/

VaultPress is a service by Automattic (monthly subscription, not free) that offers a combination of security monitoring and backup services. Definitely worth a look if your site is your livelyhood.
http://vaultpress.com/

WordPress Backup to Dropbox is a free plugin which will back up your site files and database and save them into your Dropbox account. Dropbox is a cloud storage service, which is also free to sign up for.
http://wpd2b.com/

Again, there are other backup plugins and services available, these are just some examples.

It can happen to you

It can happen to me

It can happen to everyone, eventually

-- Yes, *It Can Happen*, 90125

Anybody else besides me old enough to remember this song from the 80's?

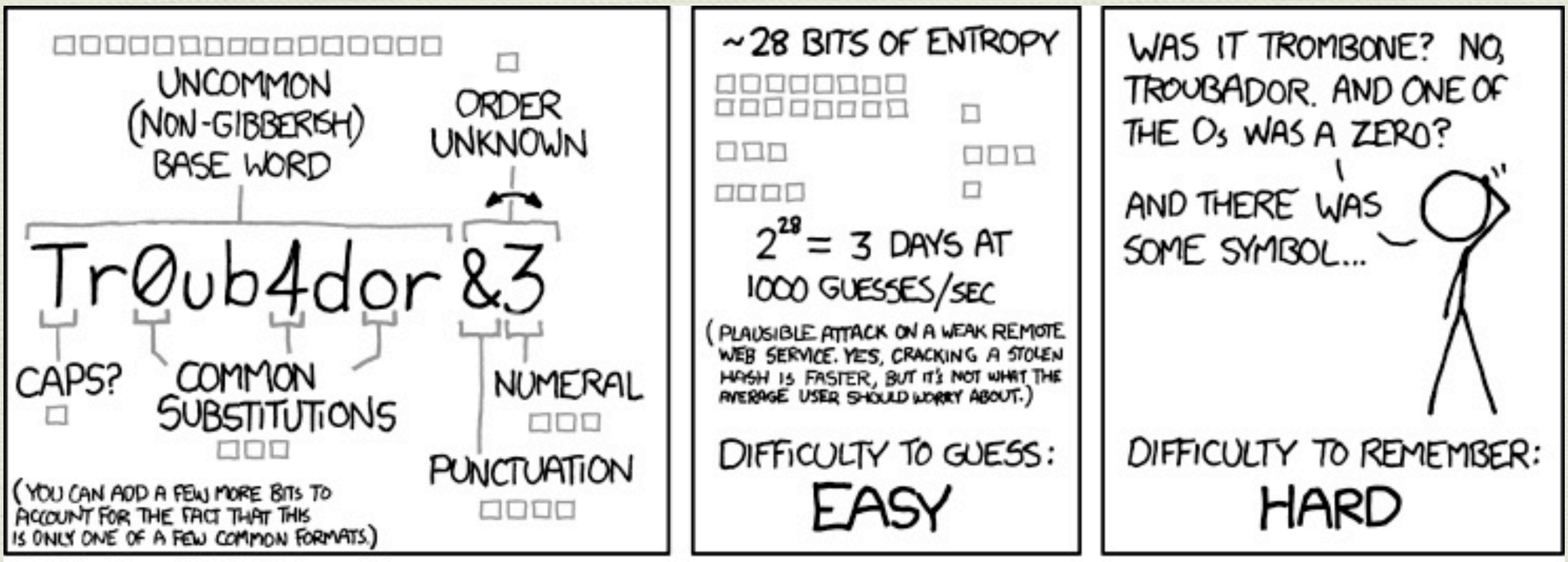
Ironically, the week before I gave this presentation at the Atlanta WordPress Users meetup, my own site was hacked. I awoke one morning to find 40 new posts on my site, hawking Viagra, Cialis, and the like. How embarassing! I found no other damage, and it appears that they had somehow obtained my password. They probably got my password by cracking it from the stolen database from some other site I used where I shared the same password. Yes, dumb mistake, and I know better.

The point is, this stuff is not just theoretical, it happens in the real world. What are the chances of it happening to you? Probably the same as the chances for me. And I got hit. What does that tell you?

A Little Healthy Paranoia

Thursday, September 27, 12

Are you paranoid yet? You need to be at least a little paranoid if you’re concerned about security.



Thursday, September 27, 12

http://xkcd.com/936/

Really good observation about password strength. A hacker could brute-force the ‘Tr0ub4dor&3’ password in 3 days (just by sending login requests to a web site over and over), or less if cracking a known hash (captured from a site password file). “Perceived complexity” of the password matters much less than the *length*. Every bit of entropy doubles the time needed to crack/guess. Exponential progression FTW!

One of my pet peeves is when a site puts limits on what you can use for your password, in a misguided attempt to force you into creating a “secure” password. How many times have you seen a site that tells you “Passwords must be 6-12 characters in length, must contain both uppercase and lowercase letters, at least one number, and at least one special character”? Every one of those constraints *weakens* the overall strength by *reducing* the total number of possible combinations that could be used.

Suggesting those options is a good idea, because passwords like that are at least not trivially guessable. Enforcing the constraints is a bad idea. A password like “correct horse battery staple”, which is 28 characters long (including spaces) and uses only lowercase letters is *much* more secure than “Tr0ub4dor&3”, which uses more types of characters but is only 10 characters long.

Ironically, banking sites are notoriously bad about this.

This is not to say that a password like “Tr0ub4dor&3” is *bad*, it’s not. Well, that particular one is a bad choice ever since xkcd used it as an example... It’s just that what we *think* is a strong password isn’t always as strong as we might think, compared to other options.

If you want some really strong passwords, consider using something like 1Password: http://agilebits.com/

Healthy Paranoia

- Use strong passwords
- Two-factor authentication -- Google Authenticator plugin
- Use separate WordPress logins for publishing day-to-day content and for site administration
- Limit who can login to your site, and what permissions they have
  - Create temporary accounts for developers, if necessary

Thursday, September 27, 12

Strong passwords are one line of defense.

Even stronger are two-factor authentication systems. What is a two-factor system? It is a system that uses two completely separate methods for verifying who you are. Typically, this is through some sort of device that you (and only you) carry with you. For example, I have an "RSA SecureID" fob on my keychain, provided to me by my employer. In order to login to our VPN from home, I have to launch the VPN client, enter a 6-10 digit PIN (which I chose) AND also enter a 6-digit number that appears on the fob. The number on the fob changes once-per-minute. So even if someone guessed my PIN, if they don't also have my keychain fob, they'd have a snowflakes chance in Hades of guessing the other half of the code in a given 1-minute window. Likewise, even if I lose my keys, if the person who has them doesn't know my PIN, they won't be able to log in.

There are similar two-factor services that work by texting a time-limited code to your cell phone, or by other call-back methods.

There is a plugin for WordPress called Google Authenticator which implements two-factor authentication via an app on your smartphone (iPhone, Android), and your GMail account. http://wordpress.org/extend/plugins/google-authenticator

There is a concept known as "separation of concerns". Do you use the same WordPress login for posting new content to your site as you do for administrative tasks like managing themes, plugins, and site settings? Why? Convenience, I'm sure. But you don't *need* all the special administrative privileges for just writing posts and pages. If you use a separate account, set to a lower access level, like the "Editor" or "Author" role, you can limit exposure of the administrative account the outside world.

If you have other users who login to your site (co-authors, guest bloggers, subscribers who can manage certain account settings, etc), make sure they have the *least* privileges that

Healthy Paranoia

- Use secure protocols: SFTP, SCP, SSH -- **not** FTP
- If possible, enforce SSL on WordPress logins and dashboard access
- Ensure MySQL server is not accessible to other hosts
- Same goes for memcache (or any other data store)

Thursday, September 27, 12

If you are using FTP to move files back and forth to your server, please stop. FTP is an old protocol written at a time when network access was very limited, and the ability to sniff passwords off of a network required very special tools and knowledge. That is no longer the case. And FTP sends your password over the network in the clear. Anybody who can sniff the network can snag your password out of an FTP transaction. Do you like to work in a coffee shop, on an un-encrypted WiFi network? Welcome to Paranoialand. That kid sitting at the corner table might not be posting funny cat pictures to Facebook. He might be snickering at how easy it is to steal passwords in a coffee shop.

Your host almost certainly supports SFTP, which is Secure FTP. Use that instead. In some cases, you might have access to SCP (Secure CoPy), or SSH, which is a secure alternative to Telnet, for logging into a terminal session (not all hosts give you that kind of access).

Back in WordPress land, you can enable settings in your wp-config.php file to force all logins or dashboard access to be via SSL (https). This will require you to install an SSL certificate on your web server. You should be able to find instructions about these topics on the WordPress Codex and via your web host's support. Administrative interfaces like CPanel and Plesk typically make it pretty easy to do this. A *real* SSL certificate, signed by a Certificate Authority, normally costs about $100/year. If you just want to protect your own login info on your site, you can create a self-signed certificate. But with those, your browser will present a warning about the trustworthiness of the certificate, so if you need SSL for something like an e-commerce site, you'll need to purchase the real thing.

Your MySQL server should be configured to *only* accept connections from your webserver and itself. In most cases, you probably have MySQL and Apache running on the same host, so just limiting connections to "localhost" will do the trick. But if you are one of the rare people (like me) who runs their database on a separate host from your web server, make sure it's not set to accept connections from any host (designated by "%" in the hostname field of your database permissions).

What? I don't know how!

That all sounds like great advice, but this technical talk about configuring servers, limiting access, database-this, and certificate-that is like Moon Language to me! How am I going to be able to do all these things? Do I have time for this? Is it worth the trouble?

Getting help

- Security is part of the cost of doing business, like insurance
- If you don't know how to do all this, retain the services of someone who does
- Managed hosting:
  - Page.ly
  - WordPress.com
  - WP Engine
  - Zippykid

Thursday, September 27, 12

First of all, you should probably be looking at site security as part of the "cost of doing business". Obviously that particularly applies if your web site *is* a business. If your site goes down, does it mean lost revenue? Then keeping it up is in your best interest!

Security is HARD! There are so many pieces to a complex system like a web application. You have the application itself (WordPress, Drupal, Ruby on Rails apps, Django apps, etc), the language platform (PHP, Perl, Python, Ruby, Java, etc), the web server (Apache, Nginx, Tomcat, etc.), the operating system (Linux, FreeBSD, Solaris, etc), and even down to the network itself (internet routers, switches, WiFi connections). I know more than the average bear about some of these things, but I would not claim to be a "Security Expert".

If your web site is mission critical, generating your income, and you don't feel like you know enough to manage security on your own, find somebody to help you. There are companies and individuals who can sell you their services to help with preventative hardening of your server, periodic monitoring, security auditing of code, or even disaster recovery. I mentioned Sucuri.net earlier, they are one such service.

Another option is to used managed hosting. With a managed host, the hosting company typically takes care of things like backups and upgrades for you (but check their terms to be sure). These are some WordPress-specific hosting options. They all have their pros and cons, which you would need to weigh before deciding who to use. There are also more general managaged hosting offerings, such as from Dreamhost and Rackspace.

Security for Developers

- Settings API, nonces, validation handlers
- Data escaping functions: esc_*()
  - esc_html()
  - esc_attr()
  - esc_sql()
  - esc_url() & esc_url_raw()
  - esc_js

If you yourself are a developer, get familiar with the APIs in WordPress that help you write secure code.

If you are creating a settings page for a plugin or theme, use the Settings API. It automatically handles several security tasks for you, protecting forms with nonces to prevent certain types of tricks, and letting you hook in a validation function to verify and sanitize all of the data being submitted in your form.

There are also specialized functions for escaping values in specific security-sensitive areas, including for when you need to hand-craft SQL database queries or pass values from WordPress PHP code into JavaScript in the browser. Familiarize yourself with these helper functions, and when and where to use them.

http://codex.wordpress.org/Settings_API
http://codex.wordpress.org/Function_Reference
http://markjaquith.wordpress.com/2009/06/12/escaping-api-updates-for-wordpress-2-8/

Now, SECURE ALL THE THINGS!

Security isn't just about good passwords, or protecting your WordPress, or configuring your server, or how much you can trust your users. It's about ALL of those things.

SECURE ALL THE THINGS!

Thanks!

Dougal Campbell
@dougal
dougal.gunters.org

I’m also on FaceBook, LinkedIn, and stuff.
“Just Google for Dougal!”