Hardening WordPress
(or, How Not To Get Hacked, Then What to Do When You Are)

Resources
Codex <http://codex.wordpress.org/Hardening_WordPress>
WPSecure <http://wpsecure.net/basics/>
healthy dose of paranoia

history
<http://en.wikipedia.org/wiki/WordPress#Vulnerabilities>

Know Thy Enemy (anatomy of an attack)
- What?
bulk password (dictionary attack)
POST /wp-login.php
POST /xmlrpc.php <https://wordpress.org/support/topic/brute-forcing-via-xmlrpc>
<http://blog.sucuri.net/2014/07/new-brute-force-attacks-exploiting-xmlrpc-in-wordpress.html>
vulnerable plugin
All-in-One SEO <http://www.tripwire.com/state-of-security/top-security-stories/all-in-one-seo-pack-wordpress-plugin-vulnerabilities/>
Contact Form 7 <http://blog.sucuri.net/2014/08/database-takeover-in-custom-contact-forms.html>
Custom Contact Forms <http://arstechnica.com/security/2014/08/critical-wordpress-plugin-bug-affects-hundreds-of-thousands-of-sites/>
MailPoet <http://arstechnica.com/security/2014/07/mass-exploit-of-wordpress-plugin-backdoors-sites-running-joomla-magento-too/>
WPTouch <http://www.zdnet.com/wordpress-plugin-vulns-affect-over-20-million-downloads-7000031703/>
vulnerable theme
TimThumb <http://arstechnica.com/security/2014/06/running-wordpress-got-webshot-enabled-turn-it-off-or-youre-toast/>
form spambot
comment spam
contact form spam
SQL injection, XSS, etc.

- When?
May 2003 WP debuts
2007-2008 WP core vulnerabilities (backdoor)
Dec 2008 WP v2.7 adds one-click update feature
2013 vulnerable plugins, targeting Top 50
2013 WP v3.7 adds auto-update for patches (X.X.n, e.g. 3.9.0 to 3.9.1)
2014 brute force attacks, targeting wp-login and XML-RPC
2014 Auttomatic acquires BruteForce

- How?
brute force
<http://codex.wordpress.org/Brute_Force_Attacks>

vulnerability scan / penetration testing
brobot l itsoknoproblembro DDOS toolkit
WPScan <http://wpscan.org/>
Flunym0us <http://code.google.com/p/flunym0us/>
WP Security Scan <http://hackertarget.com/wordpress-security-scan/>
WordPress Auditor <https://github.com/0pc0deFR/Bulk_Tools/tree/master/WordPress%20Auditor>
WordPress Sploit framework <https://github.com/0pc0deFR/wordpress-sploit-framework>

targeted
social engineering
spear phishing

- Who?
script kiddies

hacker mafia -> mafia hackers
state hackers

- Why?
Willie Sutton, "because that's where the money is."
WordPress "has grown to be the largest self-hosted blogging tool in the world, used on millions of sites and seen by tens of millions of people every day."
"WordPress was used by more than 22.0% of the top 10 million websites as of August 2013. WordPress is the most popular blogging system, at more than 60 million websites."
commonality = predictability
users infrequently update
botnets make it easy, low-risk, automated

conscription (for later use e.g. DDOS, for resale)
content manipulation (spam links, IFRAME injection e.g. fake AV scams, click selling)
steal user profiles (for spam, identity theft)
= to make $

hardening
Sun Tzu, “The opportunity to secure ourselves against defeat lies in our own hands, but the opportunity of defeating the enemy is provided by the enemy himself.”

) choose your ground
trusted web host
tested WP core version
trusted theme
necessary, vetted plugins

) stay current(-ish)
update core WordPress
update theme
update plugins
caveat: test first!

) protect your general
no 'admin' account
strong, unique password
<http://www.strongpasswordgenerator.com/>
password manager (1Password/LastPass)
Mac Keychain Password Assistant

#1 recommendation: make good SOUP (Strong, Obscure, Unique Password)

) secure the borders
file permissions (read-only except within /wp-content/uploads/)
.htaccess rules

```
# Stop spam attack logins and comments
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteCond %{REQUEST_METHOD} POST
RewriteCond %{REQUEST_URI} .(wp-comments-postlwp-login)\.php*
RewriteCond %{HTTP_REFERER} !.*yourwebsitehere.com.* [OR]
RewriteCond %{HTTP_USER_AGENT} ^$
RewriteRule (.*) http://%{REMOTE_ADDR}/$ [R=301,L]
</ifModule>

# Block WordPress xmlrpc.php requests
<Files xmlrpc.php>
order deny,allow
deny from all
</Files>
```

) guard against spies
	change DB table prefix (not wp_*)
	hide WP version display (plugin)
		// remove version info from head and feeds
		function complete_version_removal() {
		* return '';*
		}
		add_filter('the_generator', 'complete_version_removal');
	move wp-config.php (outside web root)

) prepare fallback positions
	data backups
		preferably automatic (plugin)
		including DB history (for >1 day recovery option)

responding to an attack
	contact web host (make allies)
	check & archive logs (learn the attack vector)
	block IP (plugin, web server module, firewall)
	quarantine 'bad' files for forensic review (outside web root)
	scan site files (e.g. WordFence)
	revert DB (prior to initial attack to eliminate backdoors)
	change passwords, salts (wp-config.php) <https://api.wordpress.org/secret-key/1.1/salt>

security plugins
) prevention
- All In One WP Security & Firewall <https://wordpress.org/plugins/all-in-one-wp-security-and-firewall/>
- Better WP Security -> iThemes Security <https://wordpress.org/plugins/better-wp-security/>
- BruteProtect <https://wordpress.org/plugins/bruteprotect/>
- Sucuri <https://wordpress.org/plugins/sucuri-scanner/>
	<http://www.wpbeginner.com/opinion/reasons-why-we-use-sucuri-to-improve-wordpress-security/>
- WordFence <https://wordpress.org/plugins/wordfence/>

) detection
- Exploit Scanner <https://wordpress.org/plugins/exploit-scanner/>
- TAC (Theme Authenticity Checker) <https://wordpress.org/plugins/tac/>

) comment spam prevention
- Akismet (built-in, annual fee/site) <https://wordpress.org/plugins/akismet/>
- Antispam Bee <https://wordpress.org/plugins/antispam-bee/>
- Bad Behavior <https://wordpress.org/plugins/bad-behavior/>
- Cookies for Comments <https://wordpress.org/plugins/cookies-for-comments/>
- Hashcash <https://wordpress.org/plugins/hashcash/>
- Stop Spam Comments <https://wordpress.org/plugins/stop-spam-comments/>