# WordPress Security

Nikolay Zaynelov

Annual LUG-БГ Meeting 2015

nikolay.zaynelov.com
nikolay@zaynelov.com

# Introduction

# What is WordPress

- WordPress is a free and open source content management system (CMS). It is the most widely used CMS software in the world and it powers more than 23% of the top 10 million websites, giving it an estimated 60% market share of all sites using CMS.

# Who Uses WordPress?

# License

- WordPress is licensed under the General Public License (GPLv2 or later) which provides four core freedoms, and can be considered as the WordPress "bill of rights":
  1. The freedom to run the program for any purpose.
  2. The freedom to study how the program works, and to change in such a way to make it do what you wish.
  3. The freedom to redistribute.
  4. The freedom to distribute copies of your modified versions to others.

# Release Cycle

- A release cycle usually lasts around 4 months - from the initial scoping meeting to the launch of the version.
- A release cycle follows the above pattern:
  - **Phase 1: Planning and securing team leads.**
    (...)
  - **Phase 2: Development work begins.**
    (...)
  - **Phase 3: Beta.** Betas are released, and beta-testers are asked to start reporting bugs. No more commits for new enhancements or feature requests are carried out from this phase on.
  - **Phase 4: Release Candidate.** There is a string freeze from this point on. Work is targeted on regressions and blockers only.
  - **Phase 5: Launch.** WordPress version is launched and made available in the WordPress Admin for updates.
- Based on history, a major release of WordPress happens every 6 months or so.

# Version Numbering

- A major WordPress version is dictated by the first two sequences. For example, 3.5 is a major release, as are 3.6, 3.7, or 4.0. There isn't a "WordPress 3" or "WordPress 4" and each major release is referred to by its numbering, e.g. "WordPress 3.9".

  Major releases may add new user features and developer APIs. Though typically in the software world, a "major" version means you can break backwards compatibility, WordPress strives to never break backwards compatibility. Backwards compatibility is one of the project's most important philosophies, with the aim of making updates much easier on users and developers alike.

- A minor WordPress version is dictated by the third sequence. Version 3.5.1 is a minor release, as is 3.4.2. A minor release is reserved for fixing security vulnerabilities and addressing critical bugs only. Since new versions of WordPress are released so frequently – the aim is every 4-5 months for a major release, and minor releases happen as needed – there is only a need for major and minor releases.

# Preliminary Ideas

# Security is Like an Onion

# There is Always a Risk

- Fundamentally, security *is not* about perfectly secure systems. Such a thing might well be impractical, or impossible to find and/or maintain. A secure server protects the privacy, integrity, and availability of the resources under the server administrator's control.

# Security vs. Usability

- There's a fine balance between security and usability. Sometimes locking down your site makes it secure, but it's hard to use. Sometimes making your site easier to use makes it less secure. Balance has to be found.

# In the Real World

# Layers of Security

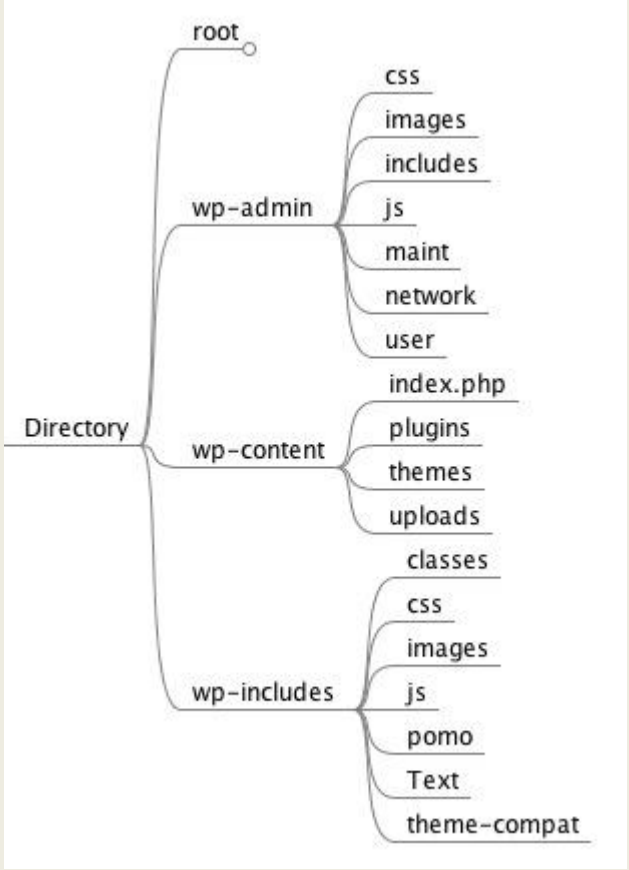
- Vulnerabilities on Your Computer
- Vulnerabilities in WordPress
- Web Server Vulnerabilities
- Network Vulnerabilities
- Passwords
- FTP
- File Permissions
- Database Security
- Securing wp-admin/ and wp-includes/
- Disable File Editing
- Security through obscurity

# Installation

- When creating user accounts for anything (server, software, databases, WordPress site, etc.) users should be granted with the minimum privilege needed to do their job
- “admin” should not be used as an account username
- Strong passwords should be required
- “wp_” table prefix should not be used

# Hardening the Installation 1/5
# File Permissions

- Owner should not be the web server user
- Files and directories should be readable by the web server user
- uploads/ should be writeable by the web server



# Hardening the Installation 2/5 Locking down wp-admin/ and wp-includes/

```
# Block wp-admin/ and wp-login.php.
<Files wp-login.php>
		Order Deny,Allow
		Deny from all
		Allow from 192.168.0.1
</Files>
<Directory /path/wp-admin/>
		Order Deny,Allow
		Deny from all
		Allow from 192.168.0.1
</Directory>

# BEGIN WordPress
```

```
# Block the include-only files.
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteBase /
RewriteRule ^wp-admin/includes/ - [F,L]
RewriteRule !^wp-includes/ - [S=3]
RewriteRule ^wp-includes/[^/]+\.php$ - [F,L]
RewriteRule ^wp-includes/js/tinymce/langs/.+\.php - [F,L]
RewriteRule ^wp-includes/theme-compat/ - [F,L]
</IfModule>


# BEGIN WordPress
```

# Hardening the Installation 3/5 Network security

- Using encryption for all traffic that contain passwords is a must

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

# Hardening the Installation 4/5 Database security

- For normal WordPress operations, such as posting blog posts, uploading media files, posting comments, creating new WordPress users and installing WordPress plugins, the MySQL database user only needs data read and data write privileges to the MySQL database; SELECT, INSERT, UPDATE and DELETE.

- Therefore any other database structure and administration privileges, such as DROP, ALTER and GRANT can be revoked. By revoking such privileges the containment policies are improving.

# Hardening the Installation 5/5 Disable File Editing

- The WordPress Dashboard by default allows administrators to edit PHP files, such as plugin and theme files. This is often the first tool an attacker will use if able to login, since it allows code execution.

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

# How to Choose a Plugin & Theme

- Download themes from well-known sources *only*
- Check out how quickly the developer responds to support requests.
- Check when the plugin is last updated.
- Check out forum threads to see how well the plugin is supported.
- Is the developer a known and respected member of the community?
- Look for a plugin that does one or two tasks really well.
- If two plugins do similar things, choose the one with the higher download count.

# Keep It Current!!!

# Security Plugins

- Login Lockdown – limits the number of failed log in attempts you can have while trying to log in
- iThemes Security
- BulletProof Security - .htaccess security protection, login security & monitoring and more…
- Exploit Scanner - searches the files on your website, and the posts and comments tables of your database for anything suspicious

# External Services

- sitecheck.sucuri.net - Free Website Malware and Security Scanner
- OSSEC - a full platform to monitor and control your systems. It mixes together all the aspects of HIDS (host-based intrusion detection), log monitoring and SIM/SIEM together in a simple, powerful and open source solution.

# Backup Strategies

1. Location - backups should be stored off-site and not on the same server as the website
2. Automatic – it's too easy for people to forget or get lazy
3. Redundancy – *"data doesn't exist unless there are at least two copies of it"*
4. Testing - make sure that the backup and restore actually work

# After the Disaster 1/2

- Take the site offline. Now. That way you avoid getting a bad rap from search engines and antivirus programs.
- Make a full backup of the infected site. It's helpful for reviewing what happened and in case you mess up something during the repair.
- Change all of your passwords and the authentication keys in the wp-config.php.
- Remove any old themes, plugins, and unused code from your server.
- Update all code on your server. Re-install WordPress so all of the WordPress files are overwritten with fresh copies.
- Reinstall themes or plugins with fresh copies to make sure no malicious code was inserted.

# After the Disaster 2/2

- Check that the file permissions on your files are correct, especially wp-config.php and uploads.
- Remove the rogue code and make sure you check all sites on your hosting account. There are tools that can help scan and clean the infection such as VaultPress. Exploit Scanner also scans for certain exploits.
- If you don't have the ability to fix the infected files the best thing to do is restore from a recent clean backup.
- Check your server access logs. Search for any bad file names that you found on your server, patterns passed as query strings, or dates/times that may clue you in to when the attack happened.
- Look closer for all POST requests before the first symptoms.
- Use system utilities like find and diff to locate the recently changed files and how the files changed.

# What Are the Most Common Attacks

- Attacks against the administrator account
- SQL Injection
- Cross-site scripting (XSS)
- Cross-site request forgery (CSRF)
- Check https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=wordpress

# Resources

1. http://codex.wordpress.org/Hardening_WordPress
2. https://github.com/WordPress/Security-White-Paper
3. https://ithemes.com/wp-content/uploads/downloads/2014/03/WordPress-Security-ebook.pdf
4. https://www.owasp.org/images/d/db/Wordpress-security-ext.pdf
5. http://build.codepoet.com/2012/07/10/locking-down-wordpress/

# Questions:

# Thank You!

Nikolay Zaynelov
E-mail: nikolay@zaynelov.com
Website: nikolay.zaynelov.com