WORDPRESS
FOR SYSADMINS

Javier Arturo Rodríguez
@codehead
Myths and Misconceptions
Myths and Misconceptions

• WordPress is:
  • Troublesome to install
  • Hard to maintain
  • Impossible to secure
NOPE
IN FACT,

- WordPress is:
  - Troublesome Trivial to install
  - Hard Easy to maintain
  - Hard to secure Can be hardened
WP-CLI
Download wp-cli

VERIFY YOUR CHECKSUMS!
Verify wp-cli

$ md5sum wp-cli.phar
8c9113f5de2a892837771fdacf6f8c16 wp-cli.phar
INSTALL WP-CLI

$ chmod +x wp-cli.phar
$ sudo mv wp-cli.phar /usr/local/bin/wp
$ wp --info
WP-CLI 0.18.0
- name: wordpress - install wp-cli
  get_url: url=https://raw.githubusercontent.com/wp-cli/builds/e94cf3fd57116b84c86a2578a0c66757fa77a592/phar/wp-cli.phar
  sha256sum=b4dd0b82df6ff3d3cbb4c9d2789dabc9f26fd21c86fc62b6f9f524d1775c9fd3
  dest=/usr/local/bin/wp-0180
  owner=root
  group=root
  mode=0755
$ wp help
https://codex.wordpress.org/Installing_WordPress#Famous_5-Minute_Install
Famous 5-Minute Install

Here's the quick version of the instructions for those who are already comfortable with performing such installations. More detailed instructions follow.

If you are not comfortable with renaming files, step 3 is optional and you can skip it as the install program will create the wp-config.php file for you.

1. Download and unzip the WordPress package if you haven't already.
2. Create a database for WordPress on your web server, as well as a MySQL user who has all privileges for accessing and modifying it.
3. (Optional) Find and rename wp-config-sample.php to wp-config.php, then edit the file (see Editing wp-config.php) and add your database information.
4. Upload the WordPress files to the desired location on your web server:
   - If you want to integrate WordPress into the root of your domain (e.g. http://example.com/), move or upload all contents of the unzipped WordPress directory (excluding the WordPress directory itself) into the root directory of your web server.
   - If you want to have your WordPress installation in its own subdirectory on your website (e.g. http://example.com/blog/), create the blog directory on your server and upload the contents of the unzipped WordPress package to the directory via FTP.
   - Note: If your FTP client has an option to convert file names to lower case, make sure it's disabled.
5. Run the WordPress installation script by accessing the URL in a web browser. This should be the URL where you uploaded the WordPress files.
   - If you installed WordPress in the root directory, you should visit: http://example.com/
   - If you installed WordPress in its own subdirectory called blog, for example, you should visit: http://example.com/blog/

That's it! WordPress should now be installed.
30-SECOND INSTALL
INSTALL WITH WP-CLI
INSTALL WITH WP-CLI

$ wp core download
$ wp core config {arguments}
$ wp core config {arguments}
$ wp user create {arguments}
OTHER WP-CLI COMMANDS

$ wp plugin install
$ wp theme install
BACKUP

$ wp export
$ wp db export
RESTORE

$ wp db import
$ wp import
Security
(FROM a SysAdmin PERSPECTIVE)
A BRUTE-FORCE ATTACK IS A BRUTE-FORCE ATTACK
Apr 21 05:40:03 myserver sshd[540]: Failed password for root from 43.255.190.187 port 40036 ssh2
Apr 21 05:40:05 myserver sshd[540]: Failed password for root from 43.255.190.187 port 40036 ssh2
Apr 21 05:40:07 myserver sshd[540]: Failed password for root from 43.255.190.187 port 40036 ssh2
Apr 21 05:40:10 myserver sshd[540]: Failed password for root from 43.255.190.187 port 40036 ssh2
Apr 21 05:40:12 myserver sshd[540]: Failed password for root from 43.255.190.187 port 40036 ssh2
Apr 21 05:40:39 myserver sshd[765]: Failed password for root from 43.255.190.126 port 47241 ssh2
Apr 21 05:40:41 myserver sshd[765]: Failed password for root from 43.255.190.126 port 47241 ssh2
Apr 21 05:40:44 myserver sshd[765]: Failed password for root from 43.255.190.126 port 47241 ssh2
Apr 21 05:40:47 myserver sshd[778]: Failed password for root from 43.255.190.126 port 37608 ssh2
Apr 21 05:40:49 myserver sshd[778]: Failed password for root from 43.255.190.126 port 37608 ssh2
Apr 21 05:40:51 myserver sshd[778]: Failed password for root from 43.255.190.126 port 37608 ssh2
Apr 21 05:40:54 myserver sshd[859]: Failed password for root from 43.255.190.126 port 54151 ssh2
Apr 21 05:47:36 myserver sshd[1937]: Failed password for root from 43.255.190.164 port 60212 ssh2
Apr 21 05:47:36 myserver sshd[1939]: Failed password for root from 43.255.190.164 port 34651 ssh2
Apr 21 05:47:39 myserver sshd[1937]: Failed password for root from 43.255.190.164 port 60212 ssh2
Apr 21 05:47:39 myserver sshd[1939]: Failed password for root from 43.255.190.164 port 34651 ssh2
Apr 21 05:47:41 myserver sshd[1937]: Failed password for root from 43.255.190.164 port 60212 ssh2
Apr 21 05:47:41 myserver sshd[1939]: Failed password for root from 43.255.190.164 port 34651 ssh2
Apr 21 05:47:44 myserver sshd[1957]: Failed password for root from 43.255.190.144 port 37130 ssh2
Apr 21 05:47:46 myserver sshd[1957]: Failed password for root from 43.255.190.144 port 37130 ssh2
Apr 21 05:47:48 myserver sshd[1957]: Failed password for root from 43.255.190.144 port 37130 ssh2
Apr 21 05:47:49 myserver sshd[1957]: Failed password for root from 43.255.190.144 port 37130 ssh2
Apr 21 05:47:49 myserver sshd[1957]: Failed password for root from 43.255.190.144 port 37130 ssh2
Apr 21 05:47:51 myserver sshd[1975]: Failed password for root from 43.255.190.144 port 52046 ssh2
Apr 21 05:47:52 myserver sshd[1975]: Failed password for root from 43.255.190.144 port 39578 ssh2
Apr 21 05:47:53 myserver sshd[1975]: Failed password for root from 43.255.190.144 port 52046 ssh2
Apr 21 05:47:54 myserver sshd[1975]: Failed password for root from 43.255.190.144 port 39578 ssh2
Apr 21 05:47:56 myserver sshd[1975]: Failed password for root from 43.255.190.144 port 52046 ssh2
Apr 21 05:47:56 myserver sshd[1975]: Failed password for root from 43.255.190.144 port 39578 ssh2
Apr 21 05:47:57 myserver sshd[1975]: Failed password for root from 43.255.190.144 port 39578 ssh2
Apr 21 05:48:19 myserver sshd[2023]: Failed password for root from 43.255.190.186 port 38812 ssh2
Apr 21 05:48:21 myserver sshd[2023]: Failed password for root from 43.255.190.186 port 38812 ssh2
Apr 21 05:48:22 myserver sshd[2023]: Failed password for root from 43.255.190.186 port 38812 ssh2
Apr 21 05:48:25 myserver sshd[2026]: Failed password for root from 43.255.190.186 port 52811 ssh2
FAIL2BAN
$ wp plugin install --activate wp-fail2ban
/VAR/LOG/AUTH.LOG

Apr 21 19:09:07 blog wordpress(my.wordpress.install)[9298]: Authentication failure for vubpuhttcp from 110.80.74.204
Apr 21 20:39:45 blog wordpress(my.wordpress.install)[9562]: Authentication failure for vubpglyghg from 110.80.74.204
Apr 21 21:11:31 blog wordpress(my.wordpress.install)[12395]: Authentication failure for mougcersdlcrh from 176.97.116.134
Apr 21 21:11:35 blog wordpress(my.wordpress.install)[8190]: Authentication failure for mouggersdvooi from 176.97.116.134
Apr 21 21:51:34 blog wordpress(my.wordpress.install)[12395]: Authentication failure for zeeidrwz54 from 199.168.141.171
Apr 21 22:05:35 blog wordpress(my.wordpress.install)[9562]: Authentication failure for carteuykiloije from 176.97.116.134
Apr 21 22:06:36 blog wordpress(my.wordpress.install)[11573]: Authentication failure for dkbzhydxmlmvj from 176.97.116.134
Apr 22 02:28:46 blog wordpress(my.wordpress.install)[12028]: Authentication failure for gerberktzksy from 176.97.116.134
Apr 22 02:28:47 blog wordpress(my.wordpress.install)[10641]: Authentication failure for carteykilohjt from 176.97.116.134
/ETC/FAIL2BAN/JAIL.Local

[wordpress]
enabled = true
filter = wordpress
logpath = /var/log/auth.log
port = http,https
bantime = 3600
# fail2ban-client status wordpress
Status for the jail: wordpress
  `- filter
    |   `- File list:/var/log/auth.log
    |   `- Currently failed: 0
    | `' Total failed: 1016
  `- action
    |   `- Currently banned: 0
    |     `- IP list:
    |     `' Total banned: 53
$ telnet wordpress
Trying 173.194.65.104...
Connected to wordpress.
Escape character is '^[].
Debian GNU/Linux 7
wordpress login:
SSH!

$ ssh wordpress
Host key fingerprint is
+-[ RSA 2048]-----+
<p>| o.+o+.         |
| ..=+ + .       |
| ..o+ E         |
|   o = + o      |
|    . . S o     |
|     . . +      |
|      . .       |
|                 |
|                 |
|                 |</p>
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
</table>

javier@wordpress password:
http://my.wordpress.com/wp-login.php
Enable HTTPS
$ a2enmod ssl
$ wp plugin install --activate wordpress-https
TWO-FACTOR AUTHENTICATION
$ wp plugin install --activate google-authenticator
Easy. Right?
Security is NOT an Add-On
CODE INJECTION
<table>
<thead>
<tr>
<th>ID</th>
<th>Added</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>6113</td>
<td>2014-08-01</td>
<td>TinyMCE 3.5 - swfupload Cross-Site Scripting Vulnerability</td>
</tr>
<tr>
<td>6114</td>
<td>2014-08-01</td>
<td>wp-3dflick-slideshow - Arbitrary File Upload Vulnerability</td>
</tr>
<tr>
<td>6115</td>
<td>2014-08-01</td>
<td>wp-homepage-slideshow - Arbitrary File Upload Vulnerability</td>
</tr>
<tr>
<td>6084</td>
<td>2014-08-01</td>
<td>RokBox &lt;= 2.13 - thumb.php src Parameter Arbitrary File Upload</td>
</tr>
<tr>
<td>6085</td>
<td>2014-08-01</td>
<td>RokIntroScroller &lt;= 1.8 - XSS,DoS,Disclosure,Upload Vulnerabilities</td>
</tr>
<tr>
<td>6086</td>
<td>2014-08-01</td>
<td>RokMicroNews &lt;= 1.5 - XSS,DoS,Disclosure,Upload Vulnerabilities</td>
</tr>
<tr>
<td>6087</td>
<td>2014-08-01</td>
<td>RokNewsPager &lt;= 1.17 - XSS,DoS,Disclosure,Upload Vulnerabilities</td>
</tr>
<tr>
<td>6088</td>
<td>2014-08-01</td>
<td>RokStories &lt;= 1.25 - XSS,DoS,Disclosure,Upload Vulnerabilities</td>
</tr>
<tr>
<td>6090</td>
<td>2014-08-01</td>
<td>sintic_gallery - Arbitrary File Upload Vulnerability</td>
</tr>
<tr>
<td>6094</td>
<td>2014-08-01</td>
<td>Shopping Cart 8.1.14 - Shell Upload, SQL Injection</td>
</tr>
<tr>
<td>6098</td>
<td>2014-08-01</td>
<td>ReFlex Gallery 1.3 - Shell Upload</td>
</tr>
<tr>
<td>6099</td>
<td>2014-08-01</td>
<td>Uploader 1.0.4 - Shell Upload</td>
</tr>
<tr>
<td>6100</td>
<td>2014-08-01</td>
<td>Uploader 1.0.4 - notify.php blog Parameter XSS</td>
</tr>
<tr>
<td>6101</td>
<td>2014-08-01</td>
<td>Uploader 1.0.0 - wp-content/plugins/uploader/views/notify.php num Parameters</td>
</tr>
<tr>
<td>6102</td>
<td>2014-08-01</td>
<td>Xerte Online 0.32 - Shell Upload</td>
</tr>
<tr>
<td>6016</td>
<td>2014-08-01</td>
<td>WordPress &lt;= 2.8.5 Unrestricted File Upload Arbitrary PHP Code Execution</td>
</tr>
<tr>
<td>5995</td>
<td>2014-08-01</td>
<td>WordPress &lt;= 3.3.2 wp-admin/media-upload.php sensitive information disclosure</td>
</tr>
<tr>
<td>5999</td>
<td>2014-08-01</td>
<td>WordPress 2.5 - 3.3.1 XSS in swfupload</td>
</tr>
<tr>
<td>5977</td>
<td>2014-08-01</td>
<td>WordPress 3.5.2 SWFUpload Content Spoofing</td>
</tr>
<tr>
<td>5982</td>
<td>2014-08-01</td>
<td>WordPress File Upload Unspecified Path Disclosure</td>
</tr>
<tr>
<td>5966</td>
<td>2014-08-01</td>
<td>WordPress Plupload Unspecified XSS</td>
</tr>
<tr>
<td>5969</td>
<td>2014-08-01</td>
<td>WordPress 3.6 SWF/EXE File Upload XSS Weakness</td>
</tr>
<tr>
<td>5972</td>
<td>2014-08-01</td>
<td>WordPress 3.6 HTML File Upload XSS Weakness</td>
</tr>
</tbody>
</table>
FILESYSTEM PERMISSIONS
# User Accounts

<table>
<thead>
<tr>
<th>Account type</th>
<th>Administrative</th>
<th>Web daemon</th>
</tr>
</thead>
<tbody>
<tr>
<td>uid</td>
<td>configmgr</td>
<td>www-data</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Permissions</th>
<th>Read/Write</th>
<th>Read-only</th>
</tr>
</thead>
</table>
## File Ownership

<table>
<thead>
<tr>
<th>Directory</th>
<th>Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>htdocs/</td>
<td>configmgr</td>
</tr>
<tr>
<td>htdocs/wp-content/uploads/</td>
<td>www-data</td>
</tr>
</tbody>
</table>
$ chown -R configmgr.wheel htdocs/
$ chown -R www-data.www-data \ htdocs/wp-content/uploads
# Prevent malicious scripts in upload dirs

```html
<Directory .../htdocs/wp-content/uploads>
  Options -Indexes -ExecCGI
  SetHandler None
  RemoveHandler .php .php3 .php4 .phps
  php_flag engine off
</Directory>
```
But this breaks auto-update!
$ wp core update
$ wp core update-db
$ wp plugin update --all
$ wp theme update --all
#!/bin/sh -
cd /var/www
wp core update
wp core update-db
wp plugin update --all
wp theme update --all
/ETC/CRON.D/WP

0 3 * * * configmgr /usr/local/bin/wp-update
SQL Injection
# apt-get install libapache2-modsecurity
# a2enmod mod-security
But I get false positives!
# Mod_security tweaks
<IfModule mod_security2.c>
  # wp-login.php should allow redirects
  <LocationMatch "^/(wp-login\./php)"
    SecRuleRemoveById 950901
  </LocationMatch>
</IfModule>
# Mod_security tweaks
<IfModule mod_security2.c>
    # wp-login.php should allow redirects
    <LocationMatch "^/(wp-login\.php)">
        SecRuleRemoveById 950901
    </LocationMatch>
    # Images are (mostly) harmless
    <FilesMatch "\.\.(gif|jpe?g|png)$">
        SecRuleEngine Off
    </FilesMatch>
</IfModule>
# Mod_security tweaks

<IfModule mod_security2.c>
  # wp-login.php should allow redirects
  <LocationMatch "^/(wp-login\.php)">
    SecRuleRemoveById 950901
  </LocationMatch>
  # Images are (mostly) harmless
  <FilesMatch "\.(gif|jpe?g|png)$">
    SecRuleEngine Off
  </FilesMatch>
  # wp-admin/ is a protected area
  <LocationMatch "^/(wp-admin/)">
    SecRuleEngine Off
  </LocationMatch>
</IfModule>

# Enable Basic Auth on /wp-admin/ to discourage attacks
<Location /wp-admin/>
  AuthType Basic
  AuthName "Restricted Area"
  AuthUserFile /etc/wpadmin.passwd
  Require valid-user
</Location>
SPAM
$ wp plugin install --activate akismet
$ wp plugin install --activate google-captcha
SetEnvIf*
.HTACCESS

SetEnvIfNoCase Via evil-spam-proxy spammer=yes
SetEnvIfNoCase Referer evil-spam-domain.com spammer=yes
SetEnvIfNoCase Referer evil-spam-keyword spammer=yes
SetEnvIfNoCase Via pinappleproxy spammer=yes
SetEnvIfNoCase Referer semalt.com spammer=yes
SetEnvIfNoCase Referer poker spammer=yes
...
Order allow,deny
Allow from all
Deny from env=spammer

https://wordpress.org/support/topic/how-to-block-semaltcom-from-visiting-your-wordpress-website
Creepy Crawlers
ROBOTS.TXT
User-agent: *
Crawl-delay: 30
MOD_QOS
MOD_QOS

# apt-get install libapache2-mod-qos
# a2enmod qos
MOD_EVASIVE
SUMMARY

• wp-cli for the win
• WordPress can be managed like any other system
• WordPress security can be managed at the system level
• Automate!
THANK YOU!

Javier Arturo Rodríguez
@codehead