OWASP

Open Web Application Security Project

Wordpress Security

GIMPA, November 2018

By: @niiankrah

What is Wordpress?

- WordPress is a free and open-source content management system based on PHP and MySQL.

- It uses a plugin architecture and a template system.

- It is most associated with blogging, but supports other types of web content including more traditional mailing lists and forums, media galleries, and online stores.

- It is also the platform of choice for over 32% of all sites across the web.

Why secure Wordpress? (1)

- Wordpress is a well developed CMS solution however no product or solution has ***absolute security***.

- Plugins and Themes might not be as secure as the base WP codebase.

- 41% of hacked WordPress were hacked through a security vulnerability on their hosting platform
- 29% were hacked via a security issue in the WordPress Theme they were using

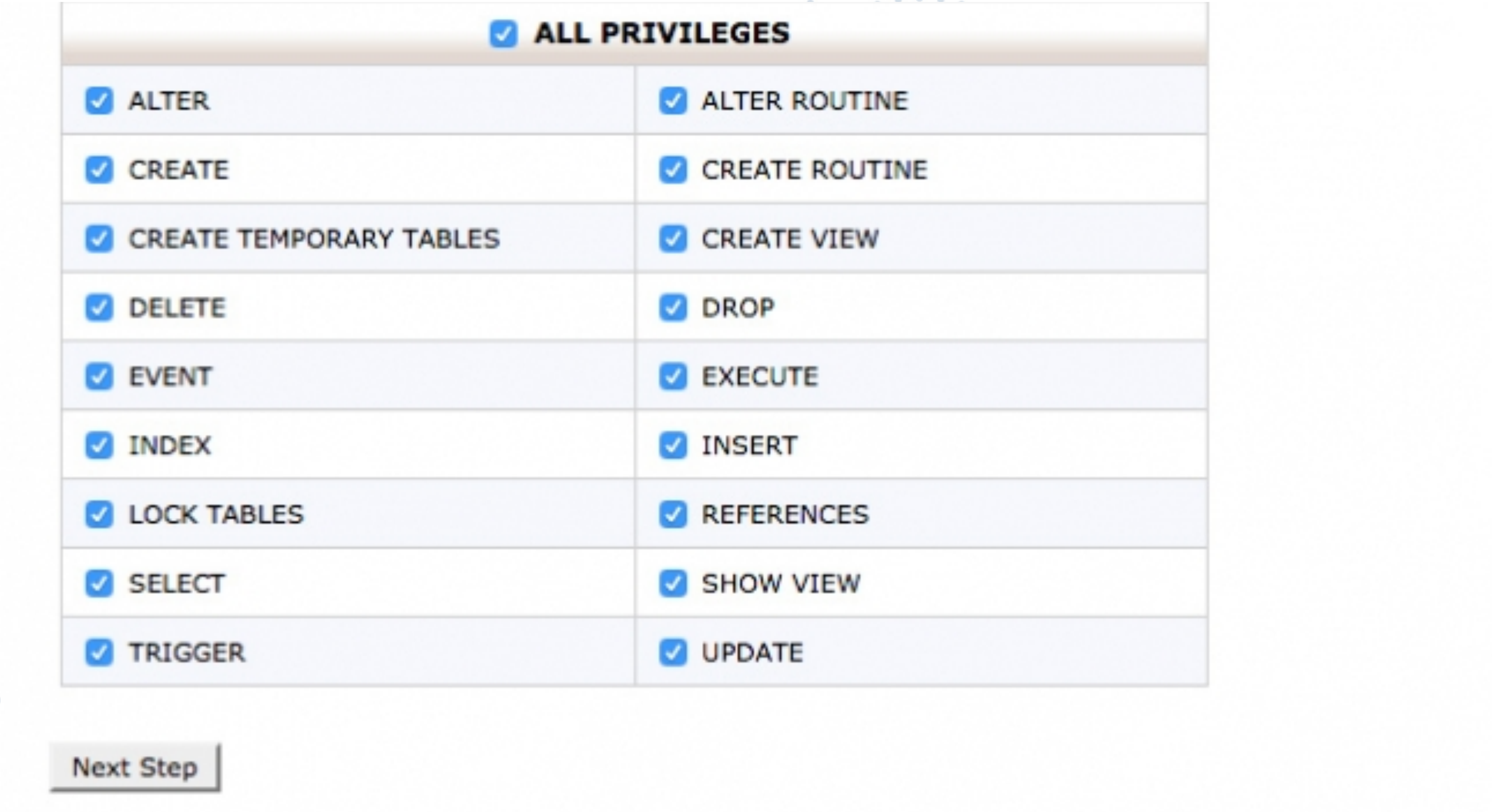
Why secure Wordpress? (2)

- 51% of hacked WordPress sites were hacked via a vulnerability in the WordPress themes and plugins they were using. (Source: wpwhitesecurity.com)

Security Concepts

- **Limit access**
- **Functional Isolation**
- **Backups**
- **Stay Up-to-Date**
- **Trusted Sources**: Do not get plugins/themes from sources that are not trusted.
- **Security Updates and News**: Security vulnerabilities is something that affects all software, WordPress is no different

Question!



☑ ALL PRIVILEGES	
☑ ALTER	☑ ALTER ROUTINE
☑ CREATE	☑ CREATE ROUTINE
☑ CREATE TEMPORARY TABLES	☑ CREATE VIEW
☑ DELETE	☑ DROP
☑ EVENT	☑ EXECUTE
☑ INDEX	☑ INSERT
☑ LOCK TABLES	☑ REFERENCES
☑ SELECT	☑ SHOW VIEW
☑ TRIGGER	☑ UPDATE

- What are the minimum DB permissions required for WP to functions?

Deployment Security - DB

- Limit database permissions
- Required permission for WP to function are SELECT, INSERT and UPDATE.
- DELETE, ALTER (for updates), CREATE TABLE, DROP TABLE require for automated updates, plug-in installation/uninstallation, etc.

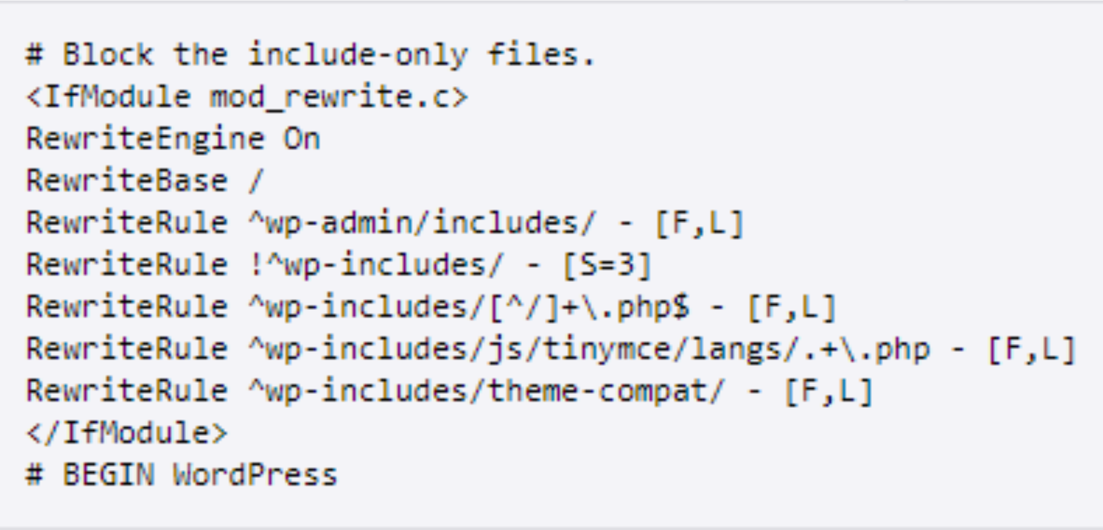
Deployment Security – Access Control

- Consider enabling 2FA by default. Some WordPress plugins designed to help include: Authy, Duo, Rublon, Two-Factor

- Make it hard for other people to guess and hard for a brute force attack to succeed. A key to this is making it Complex, Long, and Unique.

Deployment Security – WP-Includes

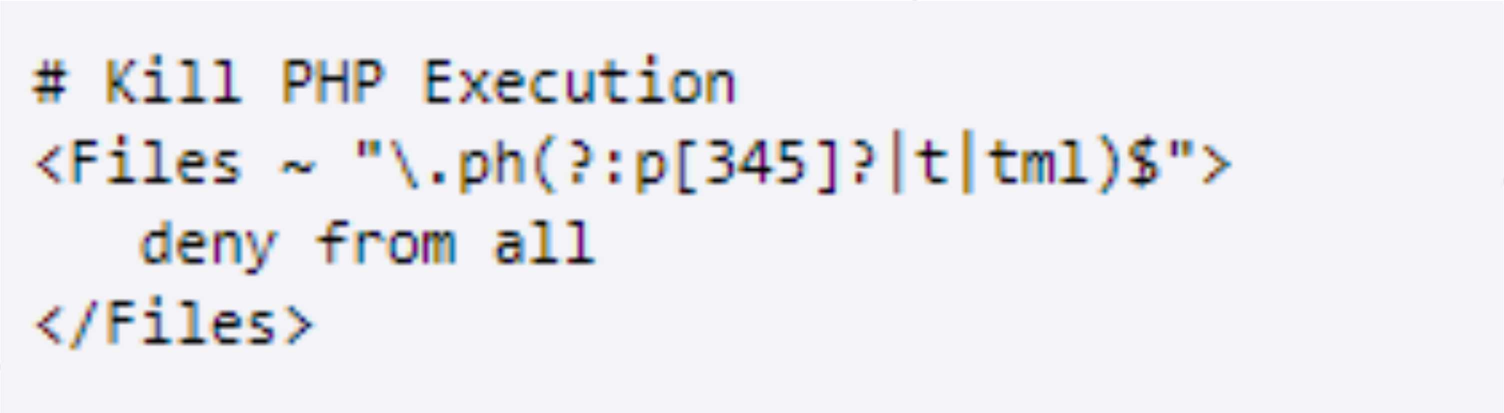
- A second layer of protection can be added where scripts are generally not intended to be accessed by any use



```
# 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
```

Deployment Security – WP-Content/Uploads

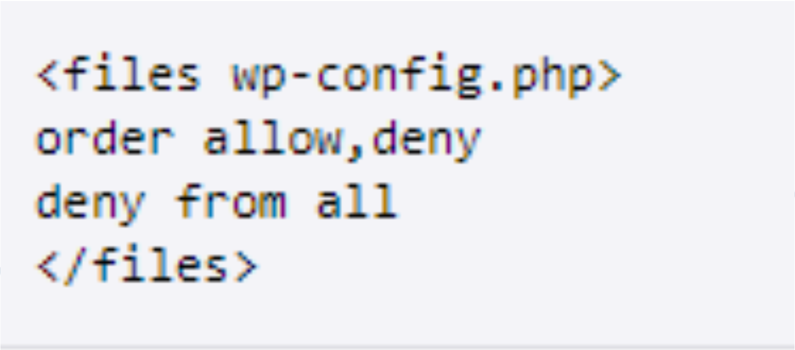
- Prevent PHP execution in this directory, you can do this by placing an .htaccess at the root of /UPLOADS using:



```
# Kill PHP Execution
<Files ~ "\.ph(?:p[345]?|t|tml)$">
   deny from all
</Files>
```

Deployment Security – Disable Editing wp-config.php

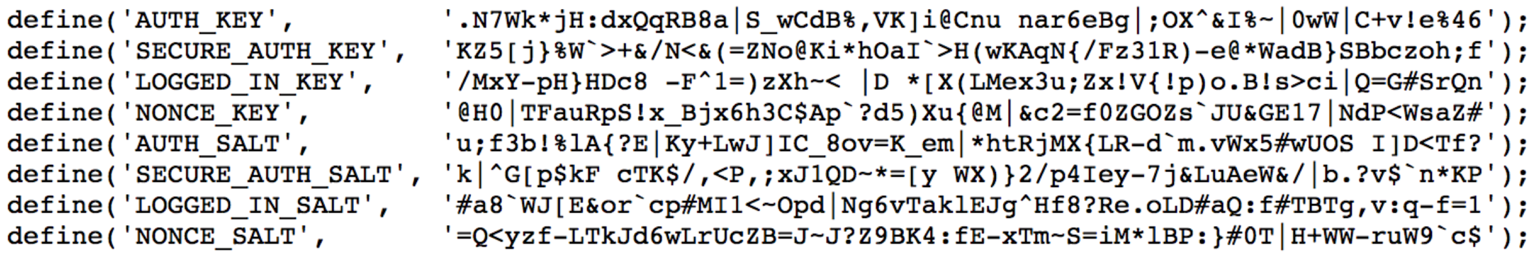
- Prevent PHP execution in this directory, you can do this by placing an .htaccess at the root of /UPLOADS using:



```
<files wp-config.php>
order allow,deny
deny from all
</files>
```

Deployment Security – Change Security Keys

- When a user logs into the Admin panel, WordPress generates cookies to keep the status of the users. To ensure that the cookies are safe and not guessable, it adds a salt while generating the cookie.



```
define('AUTH_KEY',         '.N7Wk*jH:dxQqRB8a|S_wCdB%,VK]i@Cnu nar6eBg|;OX^&I%~|0wW|C+v!e%46');
define('SECURE_AUTH_KEY',  'KZ5[j}%W`>+&/N<&(=ZNo@Ki*hOaI`>H(wKAqN{/Fz31R)-e@*WadB}SBbczoh;f');
define('LOGGED_IN_KEY',    '/MxY-pH}HDc8 -F^1=)zXh~< |D *[X(LMex3u;Zx!V{!p)o.B!s>ci|Q=G#SrQn');
define('NONCE_KEY',        '@H0|TFauRpS!x_Bjx6h3C$Ap`?d5)Xu{@M|&c2=f0ZGOZs`JU&GE17|NdP<WsaZ#');
define('AUTH_SALT',        'u;f3b!%1A{?E|Ky+LwJ]IC_8ov=K_em|*htRjMX{LR-d`m.vWx5#wUOS I]D<Tf?');
define('SECURE_AUTH_SALT', 'k|^G[p$kF cTK$/,<P,;xJ1QD~*=[y WX)}2/p4Iey-7j&LuAeW&/|b.?v$`n*KP');
define('LOGGED_IN_SALT',   '#a8`WJ[E&or`cp#MI1<~Opd|Ng6vTaklEJg^Hf8?Re.oLD#aQ:f#TBTg,v:q-f=1');
define('NONCE_SALT',       '=Q<yzf-LTkJd6wLrUcZB=J~J?Z9BK4:fE-xTm~S=iM*1BP:}#0T|H+WW-ruW9`c$');
```

- **Visit to page to generate keys**
  https://api.wordpress.org/secret-key/1.1/salt/

Deployment Security – Leverage Plugins

- There are many security plugins available for WordPress that provide a wide range of security and hardening features
- Prevention: Help protect you from hacks.
- Detection: Identify and notify if something is off and requires further inspection.
- Auditing: Track and maintain an active log of all the activity on the site (i.e., track log ins, changes to themes and plugins, updates, etc..).
- Utilities: Provide a suite of options designed to empower the user to make security-focused changes to their installation

Security through Obscurity

- There are areas in WordPress where obscuring information might help with security.
- Login page
- /wp-admin/

After Deployment Security – Continuous Monitoring

- Deploy tools that allow you to maintain visibility into the overall security state of your site.
- Examples:
- VirusTotal
- Sitecheck
- Unmaskparasites
- Redleg AW-Snap
- Quttera Web Malware Scanner
- iThemes Security

Takeaways

- Harden your WordPress after installation
- Avoid pirated themes and plugins.
- Leverage security plugins.
- Backup your website periodically
- Continuously monitor your WordPress instance

- References:

  https://codex.wordpress.org/Hardening_WordPress

  https://www.wpwhitesecurity.com/state-of-security-of-wordpress-blogs-and-websites/

Thank you!

@niiankrah

OWASP

Open Web Application Security Project