Wordpress security audit

Remediation work to be done for a hacked Wordpress website

https://patchstack.com/database/
https://blog.sucuri.net/2015/08/ask-sucuri-how-did-my-wordpress-website-get-hacked-a-tutorial.html

Is there an incident response plan?

Scan the website.

Crawlers
https://sitecheck.sucuri.net/
https://transparencyreport.google.com/safe-browsing/search
https://www.virustotal.com/gui/home/upload
Norton Safe Web
https://www.spamhaus.org/

Plugins
https://sucuri.net/wordpress-security-

plugin/

https://wordpress.org/plugins/sucuri-scanner/

https://wordpress.org/plugins/wordfence/

https://wordpress.org/plugins/tac/

Place the website in maintenance mode.

Change *all* passwords.

WordPress user accounts, WordPress hosting account, FTP or SSH user accounts, and your WordPress database password, Email accounts used for WordPress admin or hosting account.

https://haveibeenpwned.com/

https://wordpress.org/plugins/better-wp-security/

Log out all users.

https://api.wordpress.org/secret-key/1.1/salt/

Backup the website.

Reinstall Wordpress core.

Reinstall Wordpress plugins & themes.

Update plugin & themes.

Manually run the database upgrade at /wp-admin/upgrade.php.

Remove malware, hacker accounts, ..

.htaccess, wp-config.php, ..

Search your site using Google
inurl:yoursite.com viagra or cialis

Backdoors commonly include the following PHP functions

base64 str_rot13 gzuncompress eval exec system assert stripslashes preg_replace (with /e/) move_uploaded_file

Records commonly added to the wp-options table

class_generic_support widget_generic_support wp_check_hash fwp ftp_credentials

Repairing posts

```
SELECT * FROM wp_posts WHERE post_content LIKE '%<iframe%'
UNION
SELECT * FROM wp_posts WHERE post_content LIKE '%<noscript%'
UNION
SELECT * FROM wp_posts WHERE post_content LIKE '%display:%'
```

Linux commands (find & grep)

```
find .mtime -5 –ls | less
grep -ri base64 *
find uploads -name "*.php" -print
```

Rescan the website.

Check file permissions.

All your WordPress files should have 644 or 640 value as file permission. All folders on your WordPress site should have 755 or 750 as their file permission. wp-config.php permissions should be 600.

```
find /path/to/your/wordpress/install/ -type d -exec chmod 755 {} \;
find /path/to/your/wordpress/install/
```

-type f -exec chmod 644 {} \;

Remove site from Google's safe browsing list etc..

https://www.google.com/webmasters/
http://www.bing.com/toolbox/webmaster
https://webmaster.yandex.com/
https://safeweb.norton.com/tags/show?tag=WebMaster
https://www.mcafee.com/en-gb/safe-browser/mcafee-webadvisor.html

Take the website out of maintenance mode.

Hardening of a cleaned website. (wp-config.php, functions.php, site-specific plugin)

https://sucuri.net/guides/wordpress-security/
https://www.wpwhitesecurity.com/wordpress-security/
https://www.wpwhitesecurity.com/php-

hardening-wordpress/
https://patchstack.com/articles/wordpress-sensitive-information-leakage/
https://blog.sucuri.net/2012/06/how-to-lock-down-wordpress-admin-panel-with-a-dynamic-ip.html

Disable theme & plugin editors

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

Rename the administrative account

```
UPDATE wp_users SET user_login =
'newuser' WHERE user_login = 'admin'
```

Limit login attempts.

Disable PHP execution

.htaccess (wp-includes/ wp-content/uploads/)

```
<Files *.php>
deny from all
</Files>
```

Deny access to wp-config.php and .htaccess

.htaccess

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

Make sure that only you (and the web server) can read wp-config.php (it generally means a 400 or 440 permission).

https://wordpress.stackexchange.com/questions/58391/is-moving-wp-config-outside-the-web-root-really-beneficial/74972#74972

Secure wp-includes (second layer of protection)

.htaccess

```
<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
```

Password protect you admin directory.

wp-admin/.htaccess

```
#Secure Access to WP-ADMIN
ErrorDocument 401 /401.html
AuthName "Secure Area"
AuthType Basic
AuthName "Password Protected Area"
AuthUserFile /path/to/directory/.htpasswd
```

```
Require valid-user

<Files admin-ajax.php>
    Order Allow, Deny
    Allow from All
    Satisfy any
</Files>
```

wp-admin/.htpasswd

```
username:password
```

https://www.web2generators.com/apache-tools/htpasswd-generator

Limit access to wp-login.php to your IP address

/.htaccess

```
#Secure Access to WP-LOGIN.PHP by IP| Domain Name
<Files wp-login.php>
Order Deny, Allow
Deny from All
Allow from [Your IP|Your Domain Name]
</Files>
```

Limit access to wp-admin to your IP

address

wp-admin/.htaccess

```
# Secure Access to WP-ADMIN by IP|
Domain Name
<FilesMatch “.*”>
Order Deny, Allow
Deny from All
Allow from [Your IP|Your Domain Name]
</FilesMatch>
```

Disable access to XML-RPC API

.htaccess

```
#Disable Access to XML-RPC API
<Files xmlrpc.php>
Order Deny, Allow
Deny from All
</Files>
```

Disable directory indexing and browsing

.htaccess

```
#Disable Directory Indexing and Browsing
Options All -Indexes
```

Disable error reporting

wp-config.php

```
ini_set('display_errors','Off');
ini_set('error_reporting', E_ALL );
define('WP_DEBUG', false);
define('WP_DEBUG_DISPLAY', false);
```

Remove the Wordpress version number

functions.php

```
// Remove WordPress version number from
head section
remove_action('wp_head', 'wp_generator');
// Remove WordPress version number from
RSS feed
function remove_version_from_rss() {
    return '';
}
add_filter('the_generator',
'remove_version_from_rss');
```

Change the table prefix

wp-config.php

```
$table_prefix  = 'wp_123_';
```

database

```
RENAME table `wp_commentmeta` TO
`wp_123_commentmeta`;
..

SELECT * FROM `wp_123_options`
WHERE `option_name` LIKE '%wp_%'
SELECT * FROM `wp_123_usermeta`
WHERE `meta_key` LIKE '%wp_%'
```

Don’t use a robots.txt file to hide sensitive files

```
<meta name="robots" content="noindex">
```

Disable HTTP headers from the server

httpd.conf

```
Header unset Server
ServerSignature Off
ServerTokens Prod
```

Prevent cross-site scripting attacks

```
header('Content-Security-Policy: default-
src https:');
```

Thwart iframe clickjacking

```
header('X-Frame-Options: SAMEORIGIN');
```

Enable X-XSS-Protection and X-Content-Type-Options

```
header('X-XSS-Protection: 1;
mode=block');
header('X-Content-Type-Options: nosniff');
```

Add the above lines to prevent XSS attacks and tell Internet Explorer not to sniff mime types. The latter is to prevent hackers from accessing files on your server through browser functionality.

Enforce HTTPS

```
header('Strict-Transport-Security:max-
age=31536000; includeSubdomains;
preload');
```

Set Up Cookie with HTTPOnly and Secure Flag

```
@ini_set('session.cookie_httponly', true);
@ini_set('session.cookie_secure', true);
@ini_set('session.use_only_cookies', true);
```

Add the above lines to tell the browser to trust only the cookie set by the server and that the cookie is available over SSL channels.

https://securityheaders.com/

Recommendations:

Make sure the computers you use are free of spyware, malware, and virus infections. Always keep your operating system and the software on it, especially your web browser, up to date to protect you from security vulnerabilities.
https://www.malwarebytes.com/

https://www.avast.com/en-gb/index#mac

Use server-side anti-virus software.
https://www.clamav.net/

Make sure that you are sending passwords over a trusted network. An Internet cafe where you are sending passwords over an unencrypted wi-fi connection, is **not** a trusted network. Network vulnerabilities can allow passwords and other sensitive information to be intercepted.

Enforce strong passwords & implement 2FA

Keep blogs in separate databases each managed by a different user & disable remote TCP connections to the database.

Restrict database user privileges

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. Only major point upgrades (3.7 to 3.8, for example) will alter the schema.

Review log messages. Fix any PHP errors. Block IP addresses that are attempting to break into your website.

Consider installing **free SSL certificate (Let's Encrypt)** for the website, and running the website completely under HTTPS. This will allow all communications with the website to be secure; and will also help to improve your search ranking with Google.

Consider migrating your website to a **dedicated or virtual private server** (Linode?) or a **managed hosting platform** (WPEngine?) that can provide you with better security, development tools and optimised performance.

Investigate how well your website is configured for search engine optimisation; if google ranking is important for your website.

Investigate how well your website is configured for performance; again if google ranking is important for your website [Currently your website is not caching or compressing content leading to slower than necessary page load times.

Consider installing a DNS firewall & CDN (Sucuri?)

Consider installing monitoring (OSSEC?).
https://www.ossec.net/
https://perezbox.com/2013/03/ossec-for-website-security-part-i/

Register your site with the online webmaster consoles
https://www.google.com/webmasters/
http://www.bing.com/toolbox/webmaster
https://webmaster.yandex.com/
https://safeweb.norton.com/tags/show?tag=WebMaster

Linux Commands

```
find ./ -type f -mtime -15
find . -name "*.php" -exec grep
"base64"'{}'; -print &> hiddencode.txt
find /etc -type f -printf '%TY-%Tm-%Td
%TT %p\n' | sort -r .
find /etc -printf '%TY-%Tm-%Td %TT %p\n'
| sort -r .
grep --include=*.php -rn . -e
"base64_decode"
find public_html/wp-content/uploads/ -type
f -not -name "*.jpg" -not -name "*.png"
-not -name "*.gif" -not -name "*.jpeg" -not
-name “*.webp” >uploads-non-binary.log
```

```
wpscan –url http://example.com –
enumerate u
```

Assigning a MySQL user to a database

```
use mysql;
CREATE USER 'user'@'localhost'
IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON database.* TO
```

```
'user'@'localhost';
FLUSH PRIVILEGES;
DROP USER user@'localhost';
```

Updating the MySQL root password

```
sudo /etc/init.d/mysql stop
sudo mysqld_safe --skip-grant-tables &
mysql -u root
USE mysql;
UPDATE user SET
password=PASSWORD("newpassword")
WHERE user='root';
flush privileges;
quit;
sudo /etc/init.d/mysqld start
mysql -u root -p
```

Dorking Commands

```
filetype:tar.gz site:yoursite.com
filetype:sql site:yoursite.com
filetype:txt “login” site:yoursite.com
filetype:sql intext:wp_users phpmyadmin
```

```
inurl:wp-content/ “index of”
inurl:/wp-content/plugins/plugin-name/
inurl:”wordpress readme.html”
inurl:”wp readme.html”   // plugin
inurl:log -intext:log ext:log inurl:wp-

site:yoursite.com “warning” “error”
site:.*yoursite.com
```

Paul Booker

e: paul@paulbooker.co.uk
w: https://www.paulbooker.co.uk