WordPress

Security Implementation Guideline

Good practices and
epic fails of WordPress implementations

OWASP
The Open Web Application Security Project
About Me

• Information Security Consultant
  – Application Security
  – Secure SDLC

Dan VASILE

http://www.pentest.ro
dan@pentest.ro
@DanCVasile
Why do I talk about WordPress?

• I use WordPress
• Previous talk @OWASP Ro InfoSec Conf 2013
• Working with 3rd parties on secure WordPress implementation

• The project:
  WordPress Security Implementation Guideline
Why do I talk about WordPress?

WordPress Security Implementation Guideline
• Not just WordPress but Open Source adoption

• Framework for secure implementation

• Large scale integration
• General security

• Infrastructure security

• WordPress security

• Large-scale integration
• General security

• Infrastructure security

• WordPress security

• Large-scale integration
General security
- General security

- Infrastructure security

- WordPress security

- Large-scale integration
Infrastructure security
• General security

• Infrastructure security

• WordPress security

• Large-scale integration
WordPress Security Implementation Guideline

20 subjects

3 main components:
• Core
• Plugins
• Themes

Manual activities & plugin alternatives
Updates

3 main types of updates:
• Core
• Minor
• Major

WordPress > v3.7 – automatic updates for Minor
WordPress security

Updates

Turn on auto-updates for Major/Core

define( 'WP_AUTO_UPDATE_CORE', true );

For plugins and themes add a filter

add_filter( 'auto_update_plugin', '__return_true' );
add_filter( 'auto_update_theme', '__return_true' );
Choose plugins carefully

This plugin hasn’t been updated in over 2 years. It may no longer be maintained or supported and may have compatibility issues when used with more recent versions of WordPress.
Backup

What?
• *Files*
  • Core installation, plugins, themes, images & files
• *Database*

How?
Manual vs Automatic
Backup horror story 1

The good
• Daily backup, files & database, 365 days retention policy

The bad
• No geographical redundancy, no disaster recovery plan

The ugly
• HDD fail on main machine, faulty HDD on the backup machine
• Missing data and database structure
Backup horror story 2

The good

- Proper backup to the cloud

The bad

- Backup credentials stored in clear text

The ugly

- Attacker compromising site and deleting backups
User roles

- Super Admin
- Administrator
- Editor
- Author
- Contributor
- Subscriber
Restricting access

Sensitive areas of the application must be protected from unauthorized access.

.htaccess

Order Deny,Allow
Deny from all
Allow from 127.0.0.1
Write a plugin that will lock an account for a predefined period of time after a number of failed attempts.
Add blank index.php

This should be covered by Apache configuration, but it’s not always the case.
**Missing blank index.php**

<table>
<thead>
<tr>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent Directory</td>
</tr>
<tr>
<td>2011/</td>
</tr>
<tr>
<td>2012/</td>
</tr>
<tr>
<td>2013/</td>
</tr>
<tr>
<td>2014/</td>
</tr>
<tr>
<td>home-slider/</td>
</tr>
<tr>
<td>newsletter-social/</td>
</tr>
<tr>
<td>phpthumbcache/</td>
</tr>
<tr>
<td>test/</td>
</tr>
<tr>
<td>theme/</td>
</tr>
<tr>
<td>video/</td>
</tr>
</tbody>
</table>
Force encryption on data in transit

There are cases where both port 80 and 443 are used.

Sensitive operations must use SSL:

```php
define('FORCE_SSL_LOGIN', true);
define('FORCE_SSL_ADMIN', true);
```
• General security

• Infrastructure security

• WordPress security

• Large scale integration
Large scale integration

- Creating a standard image
- LDAP integration & Single Sign On
- Multisites
- Unified management of multiple installations
Creating a standard image

- Blank image (no data)
- All the updates
- All the basic shared plugins and themes (&data?)

Purpose:
- Testing ground for new stuff
- Create new instances, secure by default
LDAP integration & Single Sign On

- Integration with Active Directory
- Single Sign On (SSO)

Why?
- Centralized user management
- Use existing hierarchy
Large scale integration

Multisites

• Built-in WordPress functionality
• End users can create their own sites on demand

Downside
• Shared components (plugins)
Unified management of multiple installations

• Self-hosted and cloud solutions

Why?

• Centralized login and management
• Push updates to all instances
What’s next?

Next steps

• Contribute to the project
  Wordpress Security Implementation Guideline
• Share the knowledge

• Write secure code for WordPress
• Help others

OWASP
The Open Web Application Security Project