Hardening WordPress

(or, How Not To Get Hacked

And What To Do When You Are)

Gregory Ray
dot gray inc.
@dotgray

Resources

- **Codex.WordPress.org** / Hardening_WordPress
- **Blog.Sucuri.net** / WordPress Security
- **WPSecure.net** / Secure-wordpress
- **WPVulnDB.com** (*WPScan Vulnerability Database*)
- healthy dose of paranoia

Preparing For War

or, If Sun Tzu Ran WordPress

"The opportunity to secure ourselves against defeat lies in our own hands, but the opportunity of defeating the enemy is provided by the enemy himself."

Sun Tzu,

The Art of War

Know Thine Enemy

"If you know your enemies and know yourself,
you will not be imperiled in a hundred battles"

What?

- bulk password (dictionary attack)
  - POST /wp-login.php
  - POST /xmlrpc.php
- vulnerable plugin
  - All-in-One SEO (19m d/l)
  - SEO by Yoast (16m d/l)
  - WP Touch (5.6m d/l)
- vulnerable theme component
  - RevSlider (100k sites)
  - TimThumb
- form spambot
  - comment spam
  - contact form spam
- DDOS, SQL injection, XSS, etc.

When?

2003 *WordPress debuts*

2007-2008 WP core vulnerabilities (backdoor)

Dec 2008 *WP v2.7 adds one-click update feature*

2013 multiple vulnerable plugins, targeting Top 50

2013 *WP v3.7 adds automatic upgrades*

2014 brute force attacks, targeting wp-login and XML-RPC

2014 *Auttomatic acquires BruteProtect*

Who?

- script kiddies
- hacker mafia -> mafia hackers
- state actors

Why?

“Because that’s where the money is.”

- Willie “The Actor” Sutton,

bank robber

“I never said that...Why did I rob banks? Because I enjoyed it.”

- Willie “The Actor” Sutton,

The Memoirs of a Bank Robber

Why WordPress?

- **popularity**
  "WordPress was used by more than 23.3% of the top 10 million websites as of January 2015. WordPress is the most popular blogging system in use on the Web, at more than 60 million websites." *(Wikipedia.org)*
- **predictability**: known structure = easier to automate attacks
- **vulnerability**: multiple code pools, slow updates
- **replicability***:* botnets make it easy, low-risk, automated

Why your site?

- conscription (for later use e.g. DDOS, for botnet resale)
- content manipulation (spam links, IFRAME injection e.g. fake AV scams, click selling)
- malware hosting
- steal user profiles (for spam, identity theft)

= to make $

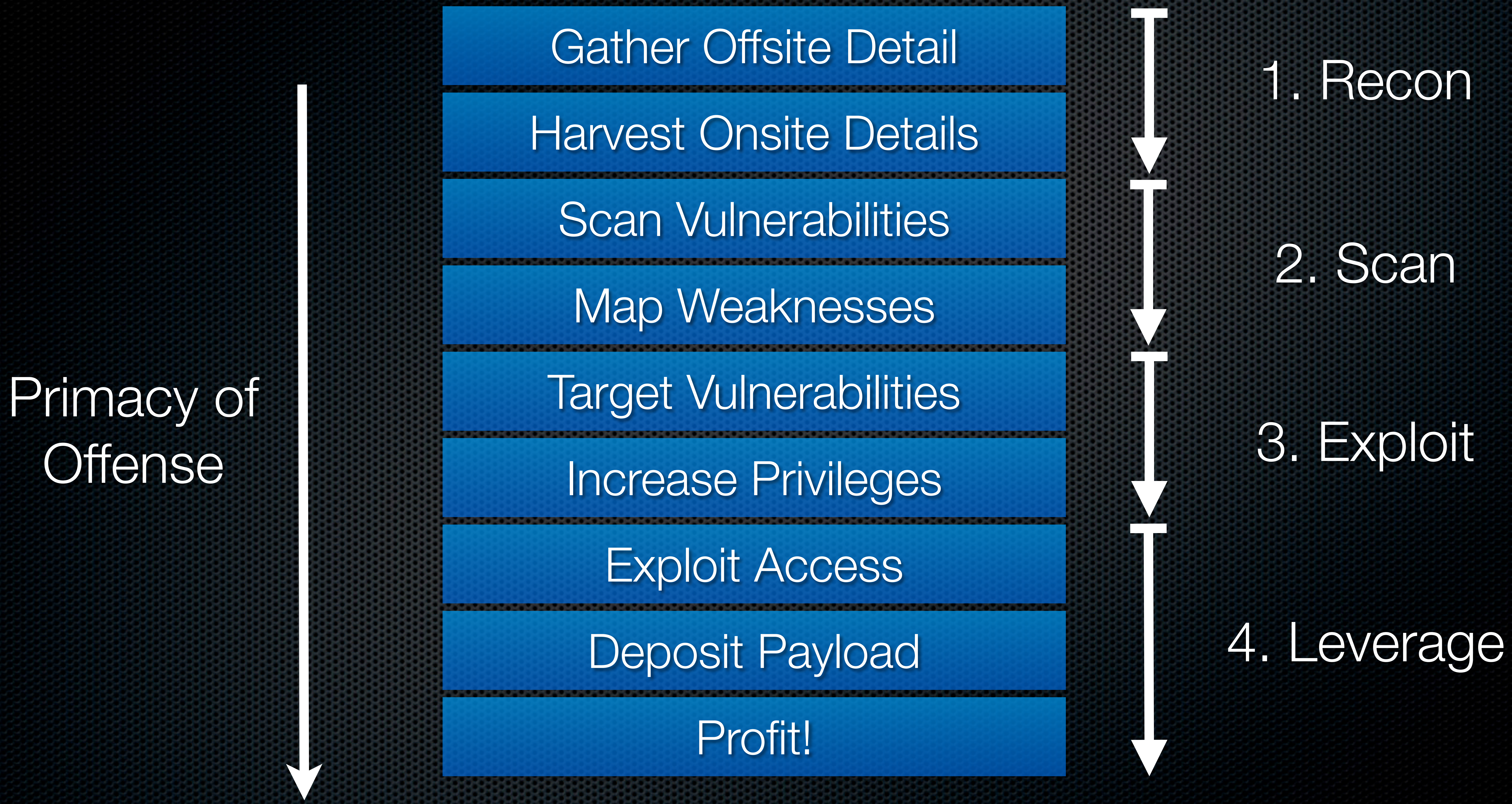
Principles Of War

or, Carrying the metaphor too far

Basic Training

1. Acquire software only from trusted sources (WP core, plugins, theme)
2. Minimize vulnerabilities by avoiding & removing unnecessary plugins
3. Stay up to date (WP core, plugins, theme)
4. Regular backups
5. Strong passwords (WP admin, MySQL, FTP)
6. Rotate keys & salts <https://api.wordpress.org/secret-key/1.1/salt>
7. No ‘admin’ account
8. Different DB prefix (not wp_*)
9. Secure access (SSL, SFTP)
10. Consider security plugins

Attack in Order



Primacy of Offense

Gather Offsite Detail

Harvest Onsite Details

Scan Vulnerabilities

Map Weaknesses

Target Vulnerabilities

Increase Privileges

Exploit Access

Deposit Payload

Profit!

1. Recon

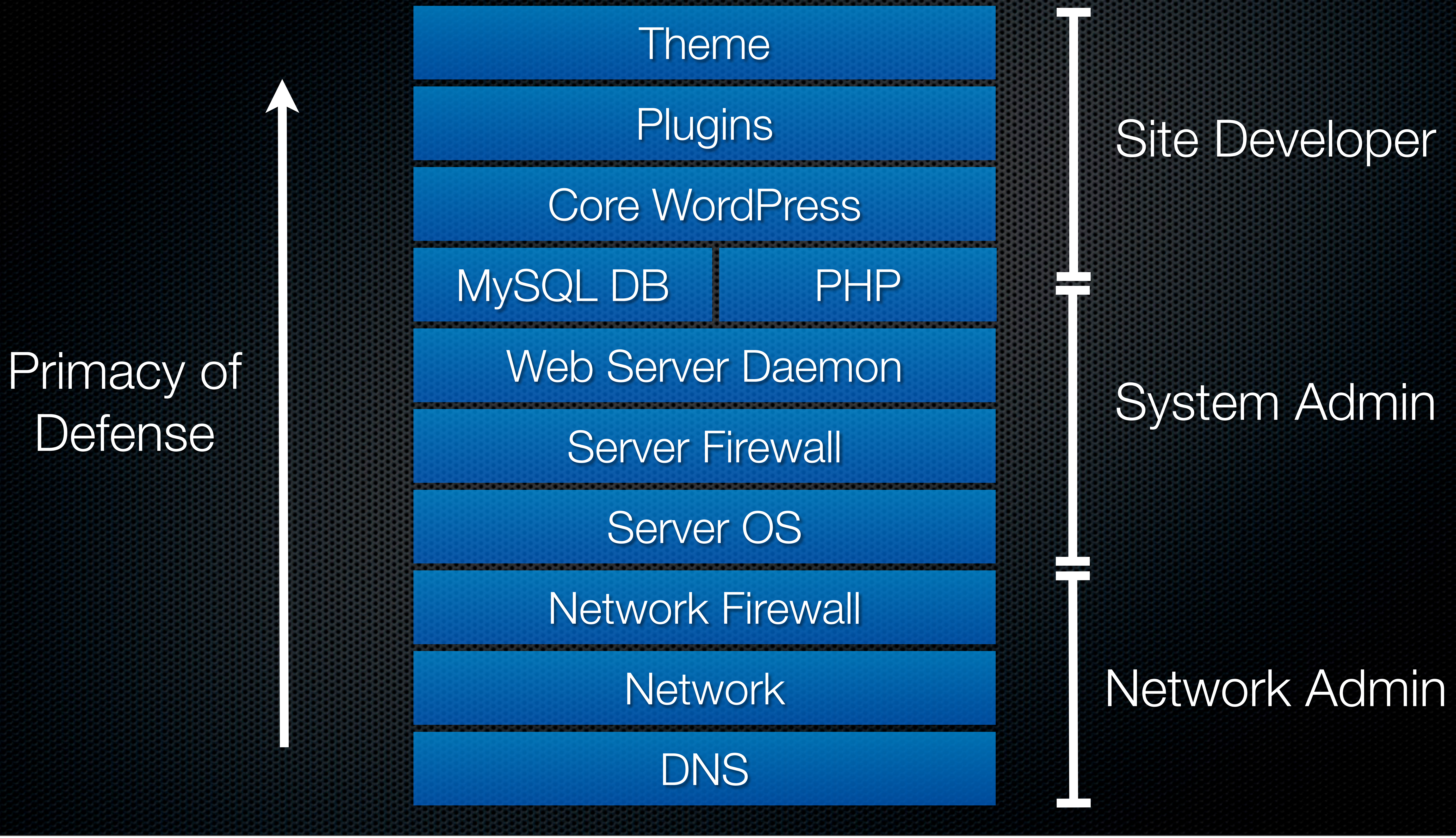
2. Scan

3. Exploit

4. Leverage

c/o anticlue.net

Defense (Vulnerability) in Depth



Primacy of Defense

Theme

Plugins

Core WordPress

MySQL DB

PHP

Web Server Daemon

Server Firewall

Server OS

Network Firewall

Network

DNS

Site Developer

System Admin

Network Admin

Security vs. Convenience

Border Security

- Database restrictions
  - Avoid multi-site unless strongly justified (shared database access)
  - Limit active user to SELECT, INSERT, UPDATE and DELETE (ALTER needed for major point releases)
- Access control
  - Basic Authentication on /wp-admin
  - Limit logins by IP
  - .htaccess (vs. bulk logins, XML-RPC, XST)
  - Plugin enforcement (iThemes Security, Wordfence)
- Disable file editing in *wp-config.php*
  - define('DISALLOW_FILE_EDIT', true);

.htaccess for dictionary attacks

```
# Stop spam attack logins and comments
<IfModule mod_rewrite.c>
 RewriteEngine On
 RewriteCond %{REQUEST_METHOD} POST
 RewriteCond %{REQUEST_URI} .(wp-comments-post|wp-login)\.php*
 RewriteCond %{HTTP_REFERER} !.*yourwebsitehere.com.* [OR]
 RewriteCond %{HTTP_USER_AGENT} ^$
 RewriteRule (.*) http://%{REMOTE_ADDR}/$ [R=301,L]
</ifModule>
```

.htaccess for XML-RPC

```
# Block WordPress xmlrpc.php requests
<Files xmlrpc.php>
order deny,allow
deny from all
</Files>
```

(can also be used for *wp-config.php*)

.htaccess for XST

```
# Disable HTTP Trace attack
RewriteEngine On
RewriteCond %{REQUEST_METHOD} ^TRACE
RewriteRule .* - [F]
```

Counter Espionage

Change what is expected, hide what is knowable.

- Block robot browsing
- Change DB table prefix (not wp_*)
- Disable WP version display (code, plugin)
- Relocate *wp-config.php* (outside web root)
- Relocate core WP files (McCreary multi-tenant method)
- Read-lock everything outside *wp-content/uploads*
  - *chmod -R 640* || *chmod -R ga-w* (depends on server user/daemon scheme)

Block robots browsing

robots.txt

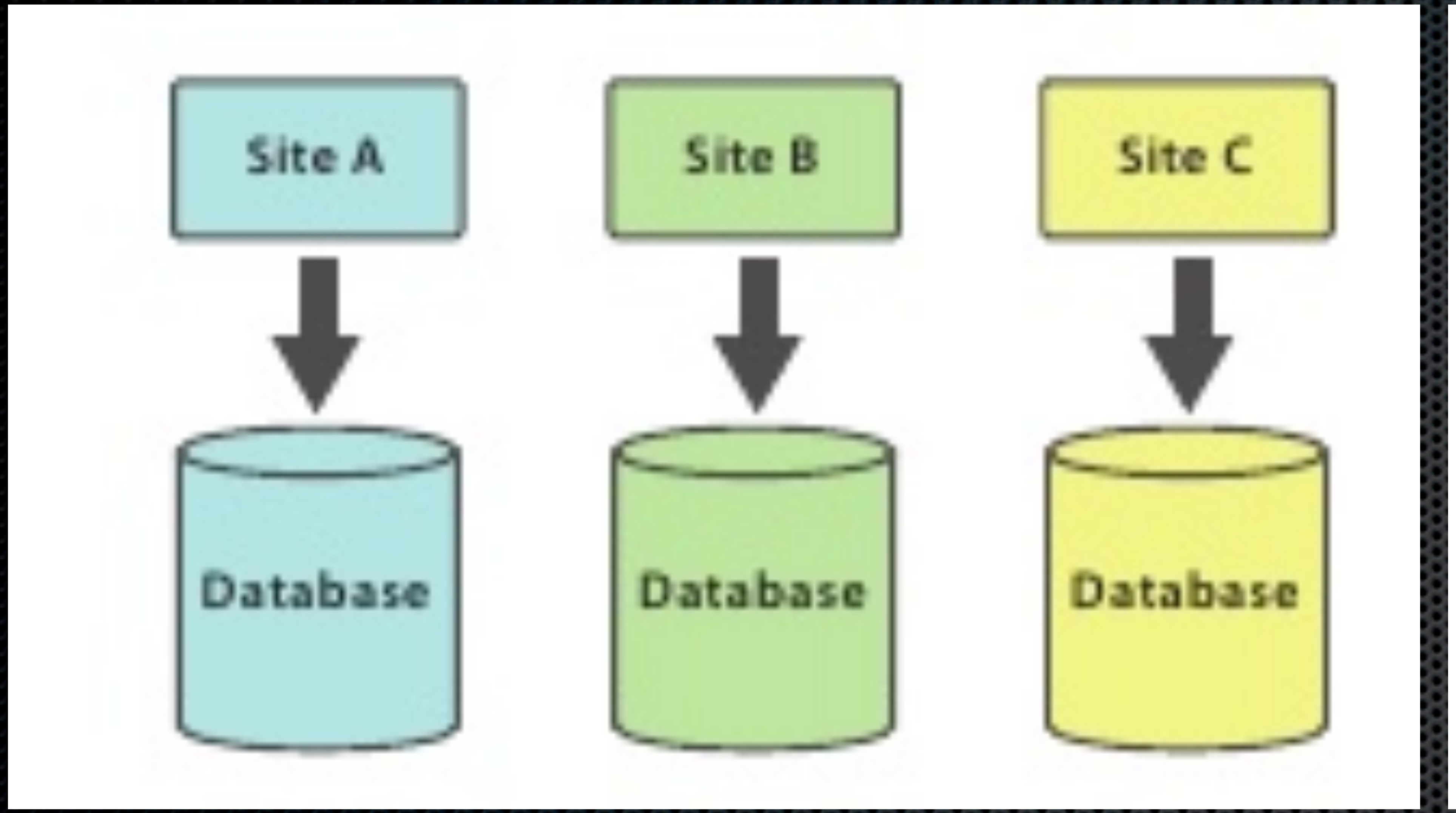
```
User-agent: *
Disallow: /wp-content/plugins/
Disallow: /wp-admin/
Disallow: /wp-content/
Disallow: /wp-includes/
Disallow: /wp-
Disallow: /xmlrpc.php
```

Disable version display

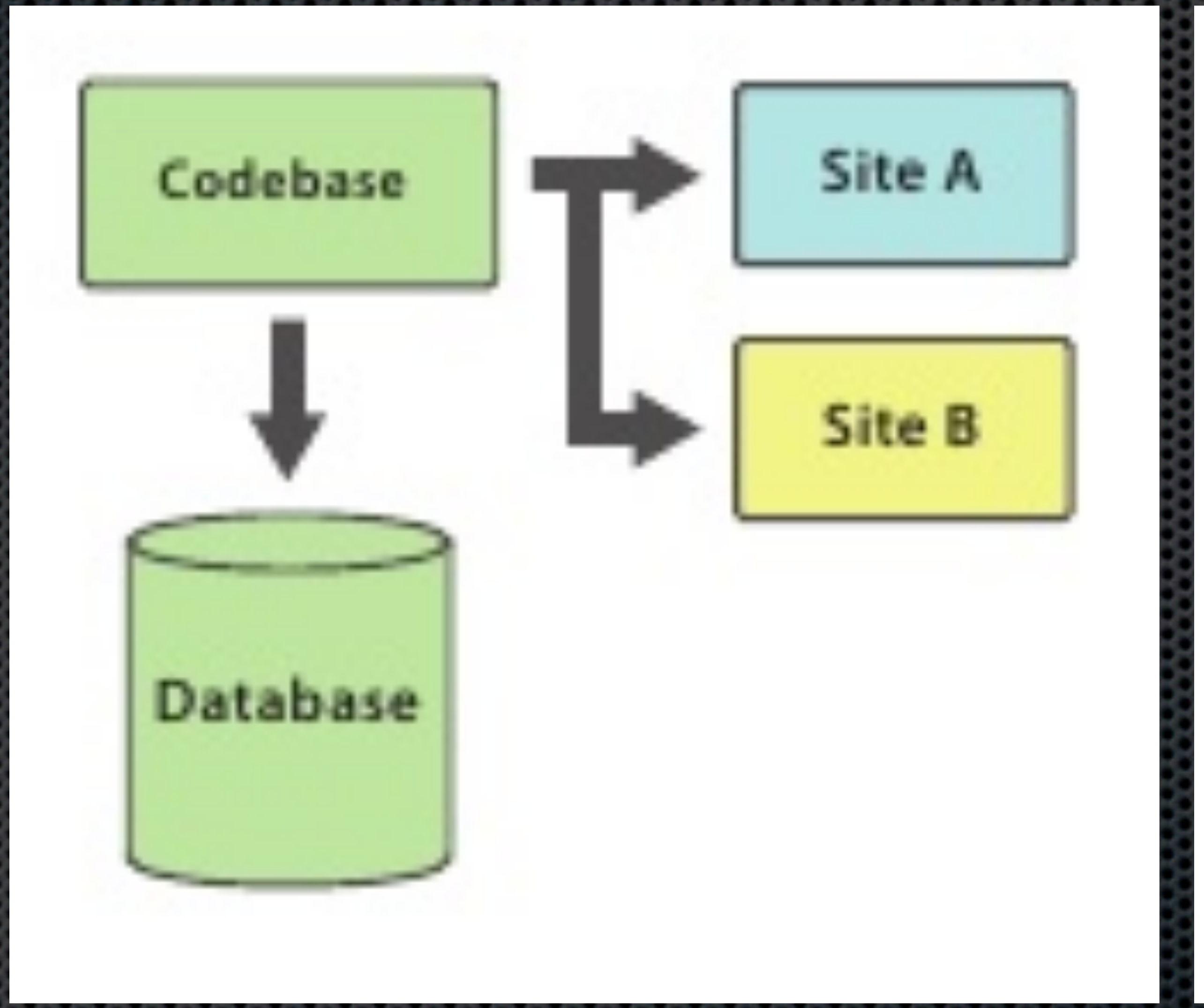
In theme’s *functions.php*:

```
// remove version info from head and feeds
function complete_version_removal() {
    return '';
}
add_filter('the_generator', 'complete_version_removal');
```

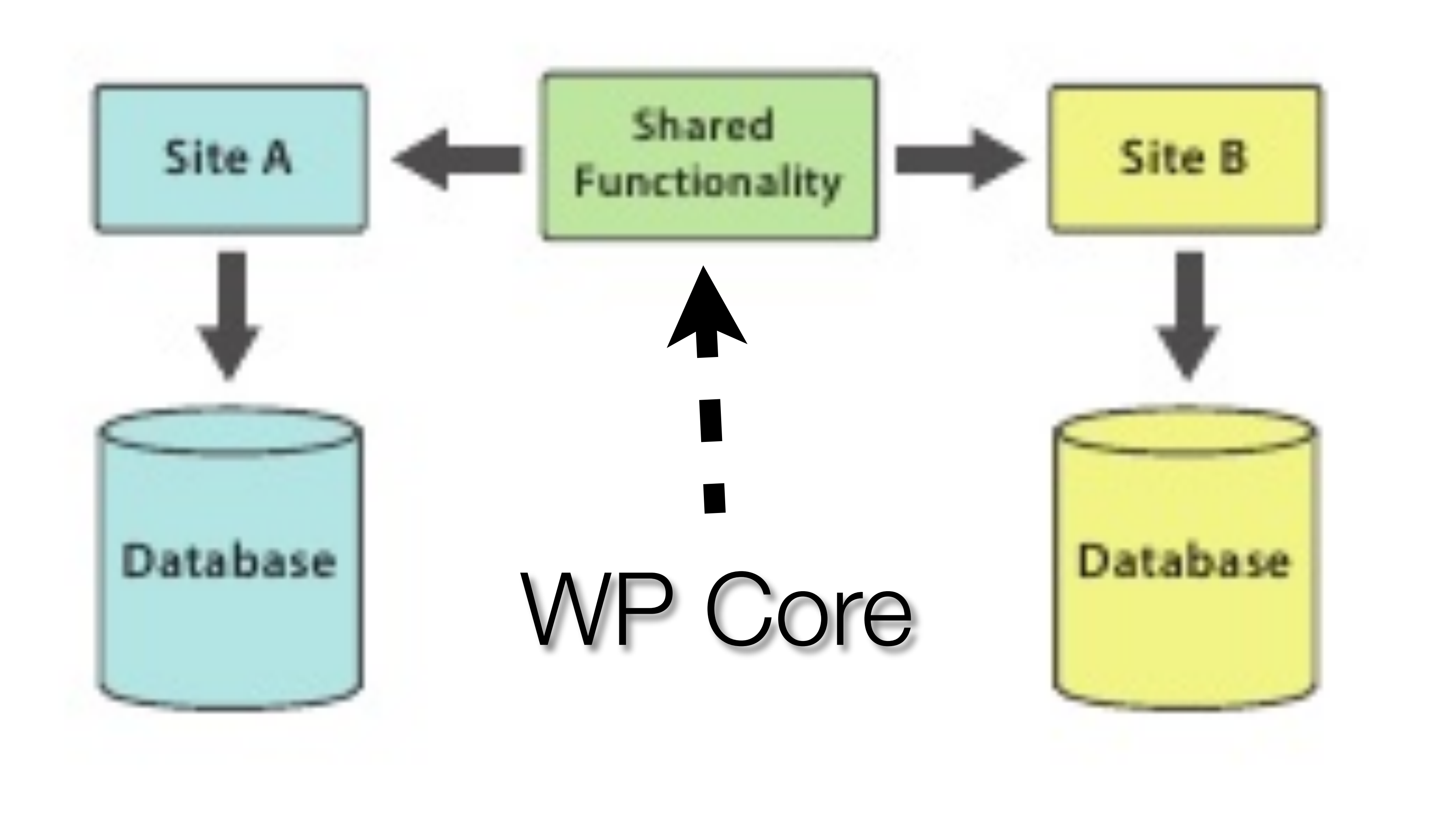
Multi-tenancy



Standalone



Multi-Site



WP Core

Multi-Tenant

Moving WP core (McCreary method)

1. Install WP into subdirectory (e.g. /core)
2. Follow ***Codex*** instructions for '*Giving WordPress Its Own Directory*'
3. Copy *wp-config.php* to site root (/)
4. Edit subdirectory *wp-config.php* to include via *$_SERVER['DOCUMENT_ROOT']*
5. Move subdirectory to core path (e.g. /usr/local/wordpress/4.0)
6. Symlink subdirectory to new core path

   ```
   ln -s /usr/local/wordpress/4.0 core
   ```

7. Site now loads *index.php*, which looks to /core/ through symlink, which references back to originating site's wp-config via *$_SERVER['DOCUMENT_ROOT']*

- 'Update' of core WordPress is now the same as 'replace symlink with pointer to different version'

  ```
  rm core; ln -s /usr/local/wordpress/4.1 core
  ```

Result of multi-tenant

```
lrwxr-xr-x  1 gray  wheel    24 Feb 18 20:42 core -> ../WPcore/wordpress-4.1/
-rw-r--r--@ 1 gray  wheel   423 Feb 17 23:28 index.php
-rw-r-----@ 1 gray  wheel  3027 Feb 18 10:19 wp-config.php
drwxr-x---  2 gray  wheel    68 Mar 13 23:17 wp-content
```

Security Drills

- vulnerability scan / penetration testing
  - brobot | itsoknoproblembro DDOS toolkit
  - Flunym0us
  - Kali
  - WPScan (.org)
  - WP Security Scan
  - WordPress Auditor
  - WordPress Sploit framework

- detection/**protection** plugins
  - **BruteProtect**
  - Exploit Scanner
  - **iThemes Security (Pro)**
  - **Sucuri**
  - TAC (Theme Authenticity Checker)
  - TimThumb Vulnerability Scanner
  - **Wordfence**

Blessed are the sysadmins

- Network-level security
  - DDOS mitigation
  - Firewall tuning
  - IDS rules
- Server-level security
  - fail2ban: protect against bulk / DDOS via IP blocking
  - mod_security: recipes to intercept attacks
  - suPHP: limit script execution by site owner (prevent neighbor attacks)
- Specialist hosting (e.g. WPEngine) and proxy/CDN (CloudFlare)

Battlefield Triage

Responding to a breach

- check for telltales
  - recent modification dates
  - Base64 encoding
- check with site host
- check & archive logs
- block IP (plugin, web server module, firewall)
- scan site files (e.g. WordFence)
- quarantine 'bad' files for forensic review
- revert DB (yay backups!)
- change passwords & salts (*wp-config.php*)

Follow Through

- Questions? Come by **Happiness Bar** next door
- Slides & speaker notes up later (check @*dotgray*)
- Extra Q&A on Sun open session (Room 301 – 2pm)