KURUKULASOORIYA KVR **200332X**

CS 3400: CYBER SECURITY PROJECT REPORT

WEEK 04 AT PANDORA COMPANY LIMITED: WEB WOES UNCOVERED

This report contains the details of observations and findings of uncovering possible vulnerabilities of the Pandora Company LTD's corporate website. The website is hosted at **http://www.pandora.lk**, is developed using WordPress, and has been operating for two years.

This report outlines 3 basic sections considered during the investigation; the current security state, possible risks and countermeasures to overcome them, and additionally best practices to maintain a good level of continuous security.

Section I: Assessment of Current State

After the discovery, the current security state of the website is identified to be at a low/moderate level and it could be concluded that the following key points are heavily contributing to the decrease in the website's security posture.

1) **Outdated WordPress Installation**

Since the website was developed and has been used for 2 years time, The WordPress core, plugins, and themes have not been updated in 2 years, which means that they may contain security vulnerabilities that can be exploited by hackers. **Starting from Sep 2021, there are close to 50 WP Core CVEs have been identified by WordPress and patches have been released to rectify them.** [1]

Thus, the WordPress installation needs to be updated to the latest version for all it's core/plugins and themes as soon as possible to mitigate the possible risks.

2) **The Uncertainty of Shared Admin Credentials**

Since the admin credentials are still sealed and not opened yet by any of the team members, it contains a huge risk in the case of an emergency, The team cannot make any changes to the website and it prevents them from fixing bugs, adding new features, or responding to security incidents.

Thus, it is mandatory to open the sealed envelope and give the admin credentials to the technical team allowing them to take control of the website in an authorized manner. Plus, it would also be a better practice to share credentials via a more secure channel than using a sealed envelope.

3) **The "Sales Inquiries" form hosted by the website**

Since this form is targeted to gather several pieces of data from potential clients, it is vulnerable to Injection attacks if any strong mechanism has been placed to validate inputs from outsiders. **SQL injection and XSS are very common types of WordPress attacks** [2], ***It is a must to implement concepts like stored procedures/ prepared statements to secure the system when queries are executed.***

4) **The utilization of MYSQL/MariaDB root user for database interaction by the website**

It is not a good practice to utilize the database root user to continue the interactions by the website because if the root user's credentials are compromised, it creates a single point of failure while leaving the entire database at risk of failing to protect CIA.

Thus, the ideal practice should be implemented to use a separate user account for WordPress with limited permissions to protect the database from unauthorized access and changes.

5) **The Website is served over HTTP rather than HTTPS**

Today is it, not a good practice to use HTTP for data transference between the server and the client because the data in transit is not encrypted and thus it is vulnerable to eavesdropping. **Plus most of search engines tend to lower such websites that use HTTP appearing in search results**[5] and this will badly affect the visibility in Search Engine Optimization.

Thus, it is recommended to use HTTPS for all data transference between the server and the client to protect the security and privacy of the users' data while enhancing their level of confidence about the website.

6) **Undefined policies against data collection/ retention and security**

At the current moment, there is no specified site-wise policy has been announced regarding how the collected data about clients via sales Inquiry form will get stored/ used and retained. **This is totally against the** General Data Protection Regulation (GDPR)**, which requires businesses to be transparent about how they collect, use, and store personal data.**

Thus, it is recommended to define a clear transparent policy on how data usage within the company, publicly announce it through the website, and collect only the data of clients who agree with the policy present. In addition to that, it would be a good practice if all the data in both retention and transit was kept encrypted.

This will protect the company against any potential legal compliance issues while enhancing its brand name's value.

Section II: Identification of Potential Vulnerabilities and Proposals for Security Interventions

In addition to the above, there are some other vulnerabilities that exist that have the ability to do potential harm to the system. These vulnerabilities can be exploited by attackers to gain unauthorized access to the system, steal data, or disrupt operations. It is important to identify and mitigate these vulnerabilities as part of a comprehensive security plan of uncovering Web Woes.

1) Brute Force Attack

This is a type of attack when an attacker involves trying an exhaustive set of combinations to guess the correct login credentials. By default, WordPress does not block a user from trying multiple failed attempts which lets a human or bot try thousands of combinations per second.[3]

In order to prevent the risk, it is suggested to;

✓ **Use a strong password policy**

This enables the users to create complex, hard-to-guess, and considerably long passwords that make them less susceptible to being cracked through exhaustive guessing.

According to the latest revision of guidelines issued by the National Institute of Standards and Technology (NIST), a U.S. federal agency, for managing digital identities it suggests that,

- User-generated passwords should be at least 8 characters long, while machine-generated passwords should be at least 6 characters in length, allowing for passwords up to 64 characters with all ASCII/Unicode characters permitted. [4]
- Passwords should be securely hashed and salted, checked against breach databases, not expire, and should not contain sequential or repeated characters. [4]
- Users should have 10 failed login attempts before being locked out, without password hints, complexity requirements, or context-specific words allowed. [4]

✓ **Use multifactor authentication**

MFA enhances the security against brute force by requiring multiple forms of verification beyond just a single password. The commonly used form of MFA is 2 Factor Authentication where the user is authenticated twice using two different types of authenticators of two different security levels which would be probably a combination of something you know/have or are. [4]

But still, NIST suggests that,

- Two-factor authentication should avoid using SMS [4]
- Knowledge-based authentication questions should be avoided. [4]

since SMS could be intercepted and Knowledge-based questions are easily guessable by the attackers. Thus, it suggests using TOTP or Authenticator apps for better security.

2) SQL Injection

Using this technique, the attacker exploits vulnerabilities in web forms or input fields to execute unauthorized SQL queries, potentially compromising the MySQL database, gaining access to WordPress admin, or altering credentials, leading to significant damage.

Consider the "Sales Inquiries Form".

The form has a field for the user's name. The attacker could enter the following in the name field:

```
' OR 1=1
```

This will inject the following SQL statement into the database:

```
SELECT * FROM sales_inquiries WHERE name = ' OR 1=1
```

The returning output will be all rows in the sales_inquiries table, regardless of the value in the name field. **Thus, the attacker can view all of the data in the table, including the names, mobile numbers, dates of birth, NICs, and home addresses of all potential clients.**

To prevent this issue, it is suggested to;

✓ Use prepared statements when writing queries to the SQL Database. **WordPress provides functions like $wpdb->prepare() to help to create secure SQL queries.**[2]

✓ Always validate and sanitize user input before using it in SQL queries. **WordPress offers functions like sanitize_text_field() and intval() to sanitize and validate input.**[2]

✓ Rather than using custom SQL queries, always try to stick to WordPress core functions that are designed with security in mind and handle SQL injection prevention.[2]

- ✓ Choose plugins wisely and always keep the WordPress core, plug-ins and themes up to date.[2]
- ✓ Use a plugin like WPScan or SucuriSiteCheck to identify whether the site has been a victim of SQL Injection or not. [3]

3) Malware

WordPress can be compromised when malicious code enters the system via sources such as infected themes, outdated plugins, or scripts. Such codes often have the capability to both extract sensitive data from the website and introduce malicious content, potentially escaping detection due to their covert characteristics.

Out of the different variations of malware, some of the most common types affecting WordPress sites include; [5]

- **Malicious redirects:** where the users are redirected to a malicious website without their knowledge or consent
- **Drive-by downloads**: where the malware gets downloaded and installed onto a user's computer without their knowledge or consent.
- **Backdoor attacks**: where a back door is created for the attacker to access the system without having to go through the normal security controls

In order to prevent this, it is suggested to;

- ✓ Choose a secure up to date WordPress plug-in like WordFence / Succuri to scan for any infections and fix them.[3]
- ✓ Download themes only from trusted resources that are free from malicious content.[3]
- ✓ Regularly keep plugins and themes up to date.[3]

4) Cross-Site Scripting

These attacks involve hackers loading pages containing insecure JavaScript scripts to steal browser data while compromising user accounts or stealing sensitive information of sessions/ cookies/ saved passwords and PINs etc.

There are 3 types of XSS approaches that WordPress sites are vulnerable to and they would be such as;

- **Reflected XSS**, where the malicious script comes from the current HTTP request, and the user is tricked into clicking and executing it.[6]
- **Stored XSS**, where the malicious script comes from the website's database.[6]

- **DOM-based XSS**, where the Document object model of the web page is manipulated to cause the page's JS to execute the attacker's code. [6]

In order to prevent this, it is suggested to;

✓ Keep the WordPress core/ plugin and themes up to date.[7]

✓ Use a strong WAF(Web Application Firewall)[7]

This can be used to block malicious traffic when it tries to enter the site.

✓ Validate and sanitize user data[7]

It is necessary to validate and sanitize all user input and data before using it in any output. **WordPress provides functions like sanitize_text_field(), esc_html(), and esc_attr() for this purpose.**

✓ Use a string CSP(Content Security Policy)[7]

By implementing a CSP in the website's headers, the execution of scripts can be restricted only to trusted sources in order to prevent inline scripts.

5) DDoS Attack

The systems become vulnerable to DDOS (Distributed Denial of Service) attacks when hackers flood servers with manipulated traffic, causing them to crash and make the website go on downtime. This mainly targets the availability feature of the website while bringing a bad reputation among the customers.

Typically these attacks are a consequence of poor-quality hosting security because attackers tend to flood the target server with traffic, making it unavailable to legitimate users. [5]

To avoid this issue, it is recommended to,[5]

✓ Monitor, detect before occurring and take preventive actions

It is recommended to use a logging feature together with the WordPress site to oversee any changes being made across your website. **There are such plugins like WordPress Activity Log that allow the site maintenance party to keep and analyze comprehensive log reports to better manage of site and easily troubleshoot.**[5]

Plus, for this purpose, a manual solution for monitoring incoming web traffic (HTTP traffic) can also be implemented as well as using yet another plugin that provides overall enhanced security such as Sucurri, WordPress Defender, etc.

✓ Use a trusted and high-quality web hosting service[5]

The first level of security against DDOS still lies with the web hosting party where it is the provider's responsibility to safeguard the server with a range of security features.

Thus, it is our responsibility to choose a reliable and trusted web hosting service provider to prevent the risk of the website being downtime due to DDOS.

6) **Open Source CMS**

Since WordPress is an Open Source Content Management System, that allows its site owners to organize, modify, and easily publish websites at their preference, **it still needs to be aware that there are some potential vulnerabilities that could arise within WordPress.**[3]

For example, **there might be some flaws within WordPress Core or Plugins that the users used**[3] Or even in authentication protocols that have been used. Sometimes, **the website managers might have not properly configured the foundation level of site security where most of the security options may still lie at its default settings.**[3]
In order to prevent such issues, it is recommended to;

- ✓ Regularly keep the WordPress core/ plugins/ themes up to date.[3]
- ✓ Install released security patches as soon as possible.[3]

However, since there is still a considerable amount of users who run their WordPress on older PHP backends, there are some compatibility issues regarding security patches released by the WordPress Team.[3] **Thus, it is a must for all users who run older PHP versions to first update the patches on the backend and then install WP security features.**

- ✓ Use a strong authentication policy and a procedure rather than simple and common which is recommended by the site settings.[3]
- ✓ Always pay attention to the initial configurations put on the configuration files.[3]

<u>Section III: Recommendations of Best Practices for Future</u>

When considering enhancing the level of Security of the WP site in the long run, the following key points could be considered important.

1. **Use SSL certificates to encrypt all data transfers between the website and the visitors.[8]**

SSL (Secure Socket Layer) SSL certificates are a type of digital certificate that verifies the identity of a website and encrypts the data that is transmitted between the website and the visitor. Adding a secure plugin to force HTTPS redirection, will enable an encrypted connection.[8]

Therefore, when someone visits the website, the web browser will verify the website's SSL certificate with the CA the issued party and if the certificate is valid only, the web browser will encrypt all data that is transmitted while securing the data transmission being private and intrinsic.

2. **Use a strong password policy and avoid the level of password reusing.**[**8**]

NIST guidelines for managing digital identities clearly suggest the required level of security in password policies and management. [4]

3. **Install a security plugin to add extra layers of protection to your website and run a security scanning tool.**[**8**]

There are many WP plugins available for enhancing security and some of the most popular would be as follows:

- **Wordfence Security:** A comprehensive security solution that offers a variety of features, including malware scanning, brute force protection, 2 Factor Auth, and a firewall with more than 4M active installations.[7]
- **Sucurri:** A cloud-based security solution that offers a variety of features, including malware scanning, website hardening, and 24/7 monitoring with close to 1M active installations.[7]
- **Jetpack:** A collection of various security features like brute force protection. File scanning and other security recommendations with more than 5M active installations.[7]
- **And AIOS / Loginizer/ SiteGuardWP and much more.**

The commonly used scanning tool by the WordPress community is

- **WPScan:** A free and open-source scanning tool that is used to scan for vulnerabilities in WP core/ themes/ plugins while maintaining a large database of known WP CVEs.[1]

4. **Keep WordPress core files, plugins, and themes updated.**[**8**]
5. **Only install plugins and themes from trusted developers.**[**8**]
6. **Run frequent backups of the website.**[**8**]

Whatever the level of security that has been placed on the web system, the rule of thumb still recommends always keeping periodic backups at an offline secure location. Because it would be the best and the most prominent adhere to protect the website from both malicious actions and human error.

7. **Don't use the "admin" username.**[8]

Once the sealed admin credentials have opened, first check for whether the admin username is “admin”. Because it poses a huge risk when most site developers use the typical “admin” as an administrator’s username. [8] **Thus, it is much better to have both strong usernames and passwords for the logins to make any unauthorized access attempts effortless.**

8. **Hide your WordPress Admin login page.**[8]

By default, most of WordPress login pages can be accessed by adding “/wp-admin” or “/wp-login.php” to the end of a URL.[8] This default behavior makes it easy for hackers to start accessing the website by first identifying the login page and then attempting on username and password. **Thus, it is a must to check for initial settings on configurations to ensure that the admin login page is hidden, and if not better to do it manually or use a plug-in like WPS Hide Login.**

9. **Disable XML-RPC.**[8]

WordPress team has implemented a protocol for XML- Remote procedure call which allows commands to be run with XML data returned. But this feature is considered to be one of the most common vulnerability features in WordPress and since most users do not need this, it is better to disable it to close the room for potential security issues.

10. **Harden your wp-config.php file (the initial configuration settings) and try to use the latest version of PHP.**[8]
11. **Choose a hosting company that takes security seriously.**[8]
12. **Use a web application firewall (WAF).**[8]

WAF security feature will protect the websites from common web application attacks by filtering out malicious traffic before it reaches your website. In WP, most of this requirement is done via plugins.

13. **Use an incidence response plan and educate the technical team.**

References

[1] WPScan, "WordPress Vulnerabilities", Available at: https://wpscan.com/wordpresses

[2]Tassos Antoniou on PRESSDIUM Blog, " SQL Injections and WordPress",[2023 May 02], Available at: https://pressidium.com/blog/sql-injections-and-wordpress/#:~:text=SQL%20injections%20are%20among%20the,types%20of%20events%20for%20you.

[3]IBAD REHMAN and WSR TEAM on WebsiteRating, " Top 7 most common WordPress Vulnerabilities (and how to fix them), [2023 September 7], Available at: https://www.websiterating.com/wordpress/most-common-wordpress-vulnerabilities/

[4]Joe Dibley on netwrix, " NIST Password Guidelines", [Published on 14 November 2022][Updated on 2023 March 17], Available at: https://blog.netwrix.com/2022/11/14/nist-password-guidelines/#:~:text=User%2Dgenerated%20passwords%20should%20be,allowed%2C%20including%20emojis%20and%20spaces.

[5]Will M on Hostinger Tutorials, " 10 Common WordPress Security Issues and How to Solve Them ", [2023 September 04], Available at: https://www.hostinger.com/tutorials/wordpress-security-issues

[6]ProtSwigger, " How to prevent XSS", Available at:https://portswigger.net/web-security/cross-site-scripting/preventing

[7]John Hudges on Themeisle, " WordPress XSS Protection: 4 Ways to Protect your site", [16 September 2021], Available at: https://themeisle.com/blog/wordpress-xss-protection/

[8]Inmotion Hosting on Search Engine Journal, " Top 15 Ways to Secure A WordPress Site, [2022 May 25], Available at: https://www.searchenginejournal.com/secure-wordpress-site-inmotion-spcs/448266/

[9]Plugins Section on WordPress, Available at: https://wordpress.org/plugins/tags/security/