Project Title: Hardening WordPress via WPScan IEEE Report

Security case project Report

Abstract

Living in a world of technology, the use of blogging on websites is only ever increasing. Resulting in a world where companies are continuously growing and creating websites that adhere to the user's need and not so much on securing the systems. Consequently, hackers are finding more vulnerabilities and exploiting those vulnerabilities on some of the most popular websites such as WordPress. The impact of this involves costing these companies Millions and Billions. This paper aims at finding vulnerabilities on WordPress via a WPScan and hardening WordPress by implementing countermeasures. Throughout this paper, there will be scans showing vulnerabilities on WordPress and research on countermeasures that has been taken to reduce the risks of cyberattacks. Countermeasures including hiding Apache version, theme in use and user accounts from a WPScan were successful. However, removing the WordPress version was a challenge of which would have been successful given more time for research and trials.

***Keywords* – WordPress, WPScan, vulnerability, cyber-attack, countermeasure, attack, HTTP, headers, Method**

Introduction

The increase use of WordPress website is continuously growing. In 2015, a research was carried out, which showed that 24% percent of the top 10 million web sites are using WordPress [24]. WordPress is a Content Management System[3] and the main usage is for blogging[10]. Many organizations have a blog, WordPress is an important web application used for blogs. As a result, attackers search for vulnerabilities in this web application, as it is a continuously growing website [3]. Research has shown that 35% of websites nowadays uses WordPress [1]. It is one of the most popular open-source website [22]. Some of the more popular CMSes, such as Joomla and WordPress [25].

It uses PHP and MySQL to create blogs to publish them dynamically[4]. PHP is a Hypertext Pre-Processor server side scripting Language used for creating webpages that are dynamic[4].

In this research paper, a research will be carried out to find out the vulnerabilities on WordPress via WPScan. The reason as to why WPScan was used is that it is the perfect environment that is a 'black box' for WordPress vulnerability scan as WPScan is for security professionals and blog maintainers to test the security of their sites [5]. Kali Linux is the tool used to conduct a WPScan to find vulnerabilities on WordPress and apply countermeasures to reduce the risks of cyber attacks. The most common place to look for vulnerabilities are plugins and themes developed by third parties [23]. The more devices a person connects, the greater the risk to the individual and to the network, and the higher the cybersecurity risk to the global infrastructure[27]. The existing cybersecurity technologies employed in distribution systems are still vulnerable to cyberattacks[26]. As a result, no system is 100% secure.

Throughout this research paper, a WPScan is carried out to discover vulnerabilities on WordPress, and countermeasures for these vulnerabilities. There will be figures showing before, during and after countermeasures are implemented to secure these systems, including screenshots to display results. These countermeasures do not mean that an attacker will not be able to attack the system. However, countermeasures do reduce the risk of the attacker attacking the system. The contents of this paper i.e. main body is separated into sections with subheadings including, HTTP header, Apache version, WordPress theme, Login details- user accounts and WordPress version.

Main body

HTTP header

In this research , HTTP 1.1 header protocol was the protocol of which was worked on. RFC standards 2616 states that HTTP is an application-level protocol [6]. HTTP is a stateless protocol used for name servers, request methods. It is an application level protocol used for distributed, collaborative, hypermedia information systems. RFC standards 2616 defines HTTP protocol, as a protocol that is based on request/response. The client sends a request method, URI and protocol version. The server responds with a status line including the message's protocol version and success or error code[6]. HTTP communication happens on TCP/IP connections[6], The default port is 80. The request header fields enables the client to give additional information about the request and information about the client itself to the

server. The server will respond with an HTTP response message after interpreting the request message from the client[6].

Methods

Method token indicates the method that need to be performed on the resource identified by the Request-URI, it is a case-sensitive method. These methods include: “GET” “HEAD” “POST” “PUT” “DELETE”. The return code of the response will let the client know if the method is allowed or not, status code of 405 means that the method is not allowed [6]. RFC standards 2616 declares the methods GET and HEAD must be supported by all general-purpose servers. All other methods are optional.

GET Method

The RFC standards 2616 identifies the GET method as a method that means retrieve whatever information (in the form of an entity) is identified by the request URI. If the request URI refers to a data-producing process, the produced data is returned as the entity in response and not the source text of the process, unless the text is the output process[6]. The use of the GET method reduces unnecessary network usage by allowing cached entities to be refreshed without requiring requests or transferring data that the client already has. The goal of caching in HTTP/1.1 is to eliminate the need to send requests and send full responses in many cases[6].

POST Method

RFC standards 2616 characterizes the GET method as a method used to request the origin server accept the entity enclosed in the request as a subordinate of the resource identified by the Request-URI in the Request-Line.

Safe Methods

RFC standards 2616, approves the POST and HEAD methods should not have significant action taken other than retrieval. These methods “ought” to be considered “safe”. This allows the user agents to represent other methods including POST, PUT and DELETE, so that the user do know that a possibly unsafe action is being requested.

Scanning WordPress, many vulnerabilities were found including the Apache version that is currently running, WordPress version, login details of users (found via enumerate users), and the theme currently in use. These vulnerabilities can be exploited by an attacker.

Figure A shows the Original WPSCAN on WordPress

```
[+] URL: http://www.honeypot.com/
[+] Started: Thu Mar  5 09:35:05 2020

Interesting Finding(s):

[+] http://www.honeypot.com/
 | Interesting Entry: Server: Apache/2.4.38 (Raspbian)
 | Found By: Headers (Passive Detection)
 | Confidence: 100%

[+] http://www.honeypot.com/xmlrpc.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
 | References:
 |  - http://codex.wordpress.org/XML-RPC_Pingback_API
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner
 |  - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access

[+] http://www.honeypot.com/readme.html
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%

[+] Upload directory has listing enabled: http://www.honeypot.com/wp-content/uploads/
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%

[+] http://www.honeypot.com/wp-cron.php
```

Figure A

```
 | Confidence: 100%

[+] Upload directory has listing enabled: http://www.honeypot.com/wp-content/uploads/
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%

[+] http://www.honeypot.com/wp-cron.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 60%
 | References:
 |  - https://www.iplocation.net/defend-wordpress-from-ddos
 |  - https://github.com/wpscanteam/wpscan/issues/1299

[+] WordPress version 5.2.5 identified (Latest, released on 2019-12-12).
 | Found By: Rss Generator (Passive Detection)
 |  - http://www.honeypot.com/?feed=rss2, <generator>https://wordpress.org/?v=5.2.5</generator>
 |  - http://www.honeypot.com/?feed=comments-rss2, <generator>https://wordpress.org/?v=5.2.5</generator>

[+] WordPress theme in use: twentyseventeen
 | Location: http://www.honeypot.com/wp-content/themes/twentyseventeen/
 | Latest Version: 2.2 (up to date)
 | Last Updated: 2020-02-25T00:00:00.000Z
 | Readme: http://www.honeypot.com/wp-content/themes/twentyseventeen/README.txt
 | Style URL: http://www.honeypot.com/wp-content/themes/twentyseventeen/style.css?ver=5.2.5
 | Style Name: Twenty Seventeen
 | Style URI: https://wordpress.org/themes/twentyseventeen/
 | Description: Twenty Seventeen brings your site to life with header video and immersive featured images. With a fo...
 | Author: the WordPress team
 | Author URI: https://wordpress.org/

 | Found By: Css Style In Homepage (Passive Detection)

 | Version: 2.2 (80% confidence)
 | Found By: Style (Passive Detection)
 |  - http://www.honeypot.com/wp-content/themes/twentyseventeen/style.css?ver=5.2.5, Match: 'Version: 2.2'

[+] Enumerating All Plugins (via Passive Methods)

[i] No plugins Found.

[+] Enumerating Config Backups (via Passive and Aggressive Methods)
 Checking Config Backups - Time: 00:00:00 <> (19 / 21) 90.47%  ETA: 00:00:0
 Checking Config Backups - Time: 00:00:00 <> (21 / 21) 100.00% Time: 00:00:
```

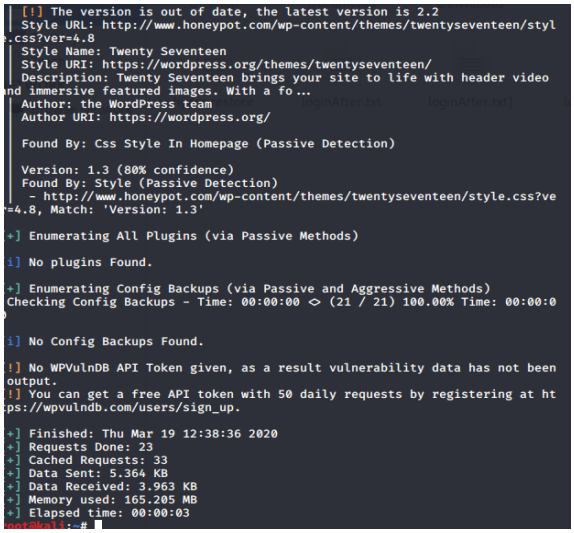


Figure A

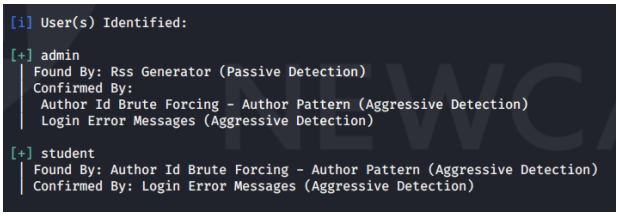


Figure A shows the list of vulnerabilities found on WordPress, such as Apache version, theme in use, WordPress version, details of user accounts

Apache version

Figure 1.1

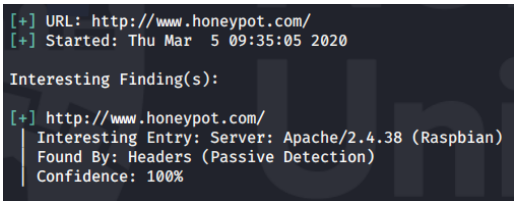


Figure 1.2

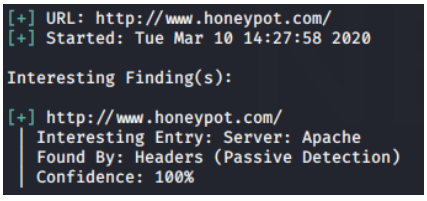


Figure 1.1 displays the Apache version which is vulnerability, This is not safe to have as an attacker can use it to exploit the vulnerabilities of WordPress through Apache version. Figure 1.2 shows after countermeasure have been implemented to hide the Apache version

Figure 1.3 below shows the codes that has been implemented in the apache configuration file (apache2.conf) to remove the header showing the apache version running.

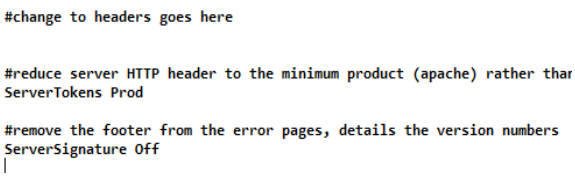


Figure 1.3

Key: # are comments

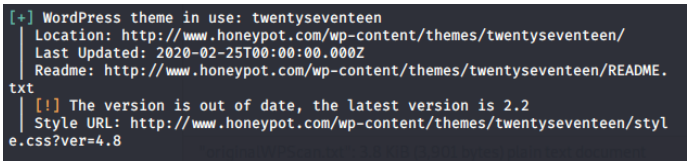
The key elements that removed the Apache version were, the **Server signature Off** and server signature identifies the web server and contains sensitive information that can be used to exploit a known vulnerability. The reason being that the version number on Apache server and Operating System will be displayed to the server signature[20]. Turning the **server signature off** means that an attacker will not know the software version of apache that is running, and will less possibly be able to exploit the software.

ServerTokens Prod, ServerTokens controls the directive of the information about the server that will be presented in server-generated documents including error messages. ServerToken sets the value of the server HTTP response header field [7].

WordPress Theme

One of the vulnerabilities discovered during the scanning of the website WordPress was the themes. The scan showed the name and type of theme that WordPress was using. This can be an exploit for an attacker as they can exploit vulnerabilities found on the theme. Resulting in having access to files and privileges of which they should not. The theme name is Twenty Seventeen, this theme's header media options allows the user to upload custom header graphics or a header video for the WordPress website[9]. The Twenty Seventeen theme comes with a default header image, many images can be used and uploaded on header image. Twenty Seventeen is activated by default.

Figure 2 shows the WordPress theme in use which is twenty seventeen



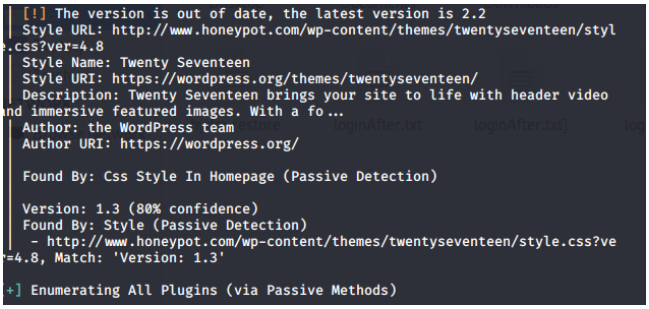


Figure 2.1

Figure 2.1 shows configuration file for Twenty Seventeen theme found on WordPress. Information such as Description, Author, Name, version, tags are displayed

Figure 2.2

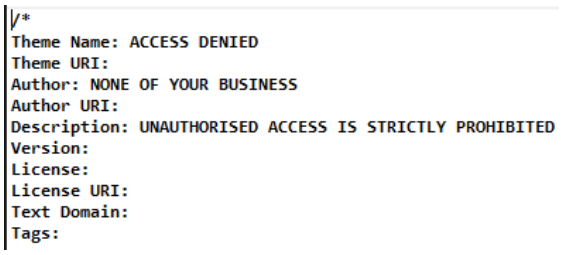


Figure 2.2 shows the changes made on WordPress styles.css for the theme by removing the information after the tags and commenting the code. Changing the configuration file style.css will display different information for the theme, this will now be displayed as above, and will prevent hackers finding out the theme in use and exploiting its vulnerabilities.

An Additional modification that was made on the theme, in order to hide it from an attacker was renaming the theme to a different name i.e. newtheme, using the command ***mv twentyseventeen newtheme***. The theme was now moved to a different location, as a result, the attacker was not able to find the theme. Figure 2.1

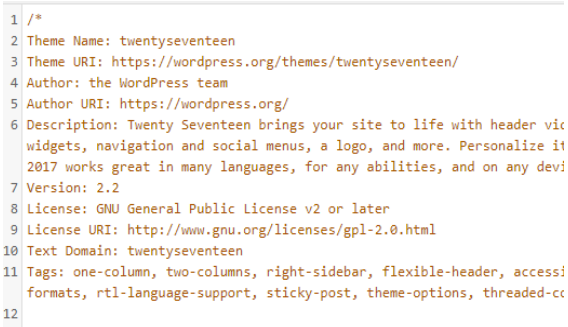
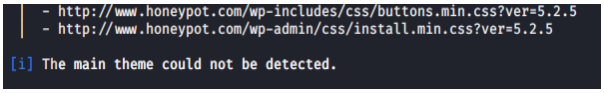


Figure 2.3 shows the results of scanning WordPress. On the bottom it is clear the main theme could no-longer be detected

Figure 2.3

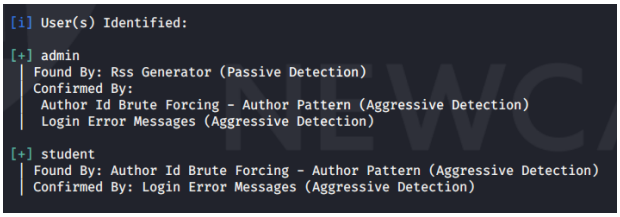


Login detail – User accounts

A New security issue that appeared on the WPScan was the login details, enumerating the users that are created on WordPress showed the list of the users including ‘student’ and ‘admin’ showing their accounts. This is dangerous from a cybersecurity perspective as the attacker can use the details of these users to get access on documents or files on the system, costing companies a lot of money. When the username on the accounts are displayed, the password could be brute forced by a hacker. This eliminate this cyber-attack, a countermeasure is to hide the usernames on the accounts.

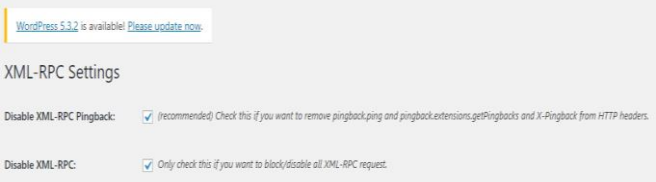
These usernames could then be brute forced by a hacker using a dictionary attack to gain access to our site. [21]

Figure 4.1 – shows the vulnerability of the user accounts ‘student’ and ‘admin’ on WordPress



XMLRPC is a system that allows remote updates to WordPress from other applications[10]. Disabling XMLRPC Allows eliminates the risk of external attacks gaining access. This reduces the risks of external attacks.

One objective was to prevent these account details being displayed on the WPScan and reduce the risk of brute force account and for a hacker not to eventually guess the correct password for these accounts [21]. Further on, there was a download of XML-RPC plugin on WordPress, to activate the plugin. After activating the plugin, disable XML_RPC.



Disabling XML_RPC on WordPress website settings

The file functions.php found on WordPress was edited including the following code:

```
function disable_wp_rest_api_error_custom($message) {
 return '';
}
add_filter('disable_wp_rest_api_error', 'disable_wp_rest_api_error_custom');
```

Figure 3 hide 'admin' user from scan

The code above, will block the 'admin' user from being displayed on the WPScan, however to block any other users the following code (Figure 3.2) will also need to be entered in functions.php

```
add_Action( 'pre_user_query', 'ap_pre_user_query' );
function ap_pre_user_query( $user_search) {
    $user = wp_get_current_user();
    if ( $user->ID !=1 ) {
    global $wpdb;
    $user_search->query_where = str_replace(
    'WHERE 1=1',
    "WHERE 1=1 AND {$wpdb->users} . ID<>1",
    $user_search->query_where
    );

    }
}
```

Figure 3.2

Figure 3 and 3.2 code used to hide user 'student' from WPScan

Figure 3.3 below

```
   - http://www.honeypot.com/wp-includes/css/buttons.min.css?ver=5.2.5
   - http://www.honeypot.com/wp-admin/css/install.min.css?ver=5.2.5

[i] The main theme could not be detected.
```

Figure 3.3 This WPScan shows results after user accounts have been hidden on WordPress. It highlights the users 'student' and 'admin' are no-longer detected on WPScan

<u>WordPress version</u>

There are multiple ways that WordPress uses to find the version. Due to this, many changes had to be made in the file functions.php. Disabling the RSS (Really Simple Syndication)feed on WordPress means that WPScan cannot detect your WordPress version, and information including the author[11]. WordPress emoji's are another way that WPScan uses to enumerate the version information, removing the emoji script being injected into the website[11] . Further on, WordPress version can be detected using meta generator tags, removing version information on the website on meta generator tags is important [11]. WPScan uses the argument **&ver=5.0.0** , this argument is added to the CSS and JS file for the WordPress version. It is useful for a caching system, changes to this will affect the caching quality of the cache, this can then remove the WordPress version[11].

WordPress Version was one of the vulnerability of which a removal of it was intended. Although, having made many changes in the configuration file on WordPress functions.php, this was not successful. Below are screenshots of what was

```
679 function remove_wordpress_version_rss() {
680     return ' ';
681 }
682 add_filter('the_generator', 'remove_wordpress_version_rss');
683
684
685 function remove_version_from_style_js( $src ){
686     if (strpos( $src, 'ver=' . get_bloginfo( 'version')))
687         $src = remove_query_arg( 'ver', $src);
688     return $src;
689 }
690 add_filter( 'style_loader_src', 'remove_version_from_style_js');
691 add_filter( 'script_loader_src', 'remove_version_from_style_js');
692
```

inserted to the configuration file functions.php on WordPress to hide its version.

Figure 6

Figure 6 – shows remove WordPress version on RSS and the header

Figure 6.2 – shows disabling RSS feed to hide WordPress version and removing unwanted headers

Figure 6.2

```
728
729 add_action('do_feed', 'shift8_security_disable_feed', 1);
730 add_action('do_feed_rdf', 'shift8_security_disable_feed', 1);
731 add_action('do_feed_rss', 'shift8_security_disable_feed', 1);
732 add_action('do_feed_rss2', 'shift8_security_disable_feed', 1);
733 add_action('do_feed_atom', 'shift8_security_disable_feed', 1);
734 add_action('do_feed_rss2_comments', 'shift8_security_disable_feed', 1);
735 add_action('do_feed_atom_comments', 'shift8_security_disable_feed', 1);
736 add_filter('the_generator', 'shift8_security_remove_wp_version_rss');
737
738 function shift8_security_remove_wp_version_rss(){
739     return '';
740 }
741 function shift8_security_disable_feed(){
742 wp_die( __( 'No feed available, please visit the <a href="'. esc_url(home_url('/')) . '">homepage</a>!'));
743 }
```

Figure 6.3

```
add_filter( 'style_loader_src', 'shift8_security_remove_wp_ver_css_js', 10, 2);
add_filter( 'script_loader_src', 'shift8_security_remove_wp_ver_css_js', 10, 2);

function shift8_Security_remove_wp_ver_css_js( $src) {
    if (strpos( $src, 'ver=' . get_bloginfo( 'version')))
        $src = remove_query_arg( 'ver', $src );
    return $src;
}


//remove wordpress version from being found in the my_footer_ssh file
function my_footer_shh() {
    if ( ! current_user_can('manage_options') ) { // 'update_core' may be more appropriate
        remove_filter( 'update_footer', 'core_update_footer' );
    }
}
add_action( 'admin_menu', 'my_footer_shh' );

```

Figure 6.2 and Figure 6.3 – shows the code used to remove WordPress version via the CSS and JS files

Etag

The RFC 2616 states that entity tags are used for comparison between two or more entities from the same requested resource. HTTP/1.1 uses entity tags in the Etag [6].

Etag are HTTP response header used to identify specific version of a resource[12]. It enables caches to be more efficient and save bandwidth, due to the fact that it does not need to be changes constantly[12]. The RFC standards 7232 describes Etag as a header filed that provides current entity-tag for selected representation in handling the request[8]. The main purpose of an entity-tag is for the service author to know the implementation of a resource, in order to select the most efficient and accurate validation mechanism for that resource[8]. .htaccess is a configuration file used for web servers that are running Apache web server software[17]. A different approach used to attempt hiding the WordPress version was disabling the Etag through configuration file of .htaccess, by modifying its configuration file[18].

Figure 5 below is the code used to ensure all Etag headers is disabled and uses cache-control headers to optimize the performance[18]. The module controls the setting of the Expires HTTP header and the max-age directive of the cache-control HTTP header in server responses. The HTTP headers uses the expiration date for instruction to the client about the validity of the document[19]. The generation of Expires and Cache-control headers can be enabled or disabled using the directive Expires Active[19]. If the .htaccess is turned off, the headers will not be generated for any document, unless overridden at a lower level such as the configuration file of a server file. If set On, the headers will be added to served documents [19]. The cache control max-age is calculated by subtracting the request time from the expiration and displaying the results in seconds[19].

```
<IfModule mod_headers.c>
                Header unset ETag
                <filesMatch "\.(ico|jpe?g|png|gif|swf)$">
                Header set Cache-Control "max-age=2592000, public"
        </filesMatch>
        <filesMatch "\.(css)$">
                Header set Cache-Control "max-age=604800, public"
        </filesMatch>
        <filesMatch "\.(js)$">
                Header set Cache-Control "max-age=216000, private"
        </filesMatch>
        <filesMatch "\.(x?html?|php)$">
                Header set Cache-Control "max-age=420, private, must-revalidate"
        </filesMatch>
</IfModule>
FileETag None
```

Figure 5

RSS feed

An RSS feed is a XML-based web content and metadata syndication format, containing basic updated information such as articles, news [13]. RSS allows authors to public notifications of new content on their site[14]. The RFC standard 4287 atom syndication format [15] is a format that describes lists of related information known as “feeds”. Feeds contain “entries” [14]. The syndication of web content such as news, weblogs are primary uses of Atom addresses. WordPress being a website used for blogging is the ‘desirable’ environment for atom syndication format which is why RSS feed is used on WordPress. At the Top level of an RSS feed is an <rss> element alongside with a mandatory attribute known as ‘version’. This specifies the version of RSS that the document is using. The <rss> element contains information about the channel i.e. the metadata and its contents[16].

Despite having done many different approaches to remove WordPress version, the version was continuously showing up. The reason being, was the version was found in several different places. Upon the removal of the Etag, WordPress used other methods such as the RSS feed to get the version. Due to the fact there were more than 3 layers – Etag , .htaccess, RSS feed, and emoji , footer, meta generated tags, unwanted header [11], all are some ways WordPress uses to find its version. Given more time, there would have been a countermeasure to successfully remove the WordPress version.

Conclusion:

In conclusion, the use of blogging websites can increase the audience on an online platform. Although there is a continuous growth of WordPress, this does not mean that these sites are 100% secure. It is important to have websites that are easy to use. Most importantly not forgetting

securing the user's data as it is always vulnerable to cyber-attack and can be exploited effortlessly. In this paper, WPScan was used to scan WordPress for vulnerabilities. This paper shows the type of vulnerabilities found on WordPress, what code was implemented for that software in order to provide countermeasures for that vulnerability. The Application of countermeasures on vulnerabilities such as Apache version, theme in use, and user accounts were successful. Though, hiding WordPress version was not successful, given more time, WordPress version countermeasure would have been successful with further research and trials. Countermeasures do not mean that a system will not be hacked, what it does mean that it will take some time before the system is hacked. As a result, reducing the risk of cyber-attacks in the near future.

References

[1]"Deep Look Into the WordPress Market Share (2019)", *Kinsta Managed WordPress Hosting*. [Online]. Available: https://kinsta.com/WordPress-market-share/. [Accessed: 08- Mar- 2020].

[2]"Usage Statistics and Market Share of WordPress, March 2020", *W3techs.com*. [Online]. Available: https://w3techs.com/technologies/details/cm-WordPress. [Accessed: 03- Mar- 2020].

[3]R. Ratnayake, *WordPress development quick start guide : build beautiful and dynamic websites for your domain from scratch*. Packt Publishing Ltd, 2018, pp. 8, 28, 57, 119, 122.

[4]L. Sabin-Wilson, *WordPress All-In-One for Dummies*, 2nd ed. Newark: John Wiley & Sons, 2013, 2013, p. chapter 1.

[5]"WPScan a WordPress Vulnerability Scanner", *Wpscan.org*. [Online]. Available: https://wpscan.org/. [Accessed: 10- Mar- 2020].

[6]"RFC 2616 - Hypertext Transfer Protocol -- HTTP/1.1", *Tools.ietf.org*, 1999. [Online]. Available: https://tools.ietf.org/html/rfc2616. [Accessed: 11- Mar- 2020].

[7]"Server-Wide Configuration - Apache HTTP Server Version 2.4", *Httpd.apache.org*, 2020. [Online]. Available: https://httpd.apache.org/docs/2.4/server-wide.html. [Accessed: 09- Mar- 2020].

[8]"RFC 7232 - Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests", *Tools.ietf.org*, 2014. [Online]. Available: https://tools.ietf.org/html/rfc7232. [Accessed: 17- Mar- 2020].

[9]L. Wilson, *WordPress For Dummies*, 8th ed. John Wiley & Sons, 2017, 2017, pp. 194, 260, 96.

[10]C. Racicot, "How to Enable and Disable XMLRPC.PHP in WordPress and Why - GreenGeeks", *GreenGeeks*. [Online]. Available: https://www.greengeeks.com/tutorials/article/how-to-enable-and-disable-xmlrpc-php-in-wordpress-and-why/. [Accessed: 20- Mar- 2020].

[11]"How to block your Wordpress site from being scanned by WPScan with Nginx - Toronto Web Design and Development | Shift8 Web", *Toronto Web Design and Development | Shift8 Web*. [Online]. Available: https://www.shift8web.ca/2019/01/how-to-block-your-wordpress-site-from-being-scanned-by-wpscan-with-nginx/. [Accessed: 23- Mar- 2020].

[12]"ETag", *MDN Web Docs*, 2019. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/ETag. [Accessed: 16- Mar- 2020].

[13]T. Lacoma, "Confused about RSS? Don't be. Here's what it is and how to use it", *Digital Trends*, 2019. [Online]. Available: https://www.digitaltrends.com/computing/what-is-an-rss-feed/. [Accessed: 18- Mar- 2020].

[14]C. Teske, "Find RSS Feeds for Your Favorite Websites and Make Life Easier", *Lifewire*, 2020. [Online]. Available: https://www.lifewire.com/what-is-an-rss-feed-4684568. [Accessed: 23- Mar- 2020].

[15]"RFC 4287 - The Atom Syndication Format", *Tools.ietf.org*, 2005. [Online]. Available: https://tools.ietf.org/html/rfc4287. [Accessed: 03- Mar- 2020].

[16]D. Winer, "RSS 2.0 Specification (RSS 2.0 at Harvard Law)", *Cyber.harvard.edu*, 2003. [Online]. Available: https://cyber.harvard.edu/rss/rss.html. [Accessed: 02- Mar- 2020].

[17]"What is .htaccess? - Apache .htaccess Guide, Tutorials & Examples", *Htaccess-guide.com*. [Online]. Available: http://www.htaccess-guide.com/. [Accessed: 13- Mar- 2020].

[18]J. Starr, "Disable ETags | .htaccess made easy", *.htaccess made easy*, 2017. [Online].

Available: https://htaccessbook.com/disable-etags/. [Accessed: 19- Feb- 2020].

[19]"mod_expires - Apache HTTP Server Version 2.4", *Httpd.apache.org*. [Online]. Available: http://httpd.apache.org/docs/current/mod/mod_expires.html. [Accessed: 22- Mar- 2020].

[20]S. K, "How to Disable Server Signature by Editing Htaccess/Apache", *TechNumero*, 2017. [Online]. Available: https://technumero.com/disable-server-signature-by-editing-htaccess-apache/. [Accessed: 24- Mar- 2020].

[21]"How to Implement Validation for Restful Services with Spring", *Knowledge Base by phoenixNAP*, 2018. [Online]. Available: https://phoenixnap.com/kb/prevent-brute-force-attacks. [Accessed: 17- Mar- 2020].

[22] M. Tomiša, M. Milković and M. Čačić, "Performance Evaluation of Dynamic and Static WordPress-based Websites," 2019 23rd International Computer Science and Engineering Conference (ICSEC), Phuket, Thailand, 2019, pp. 321-324.

[23] I. Cernica and N. Popescu, "Wordpress Honeypot Module," 2018 IEEE 16th International Conference on Embedded and Ubiquitous Computing (EUC), Bucharest, 2018, pp. 9-13.

[24] I. Cernica, N. Popescu and B. ţigănoaia, "Security Evaluation of Wordpress Backup Plugins," 2019 22nd International Conference on Control Systems and Computer Science (CSCS), Bucharest, Romania, 2019, pp. 312-316.

[25] M. Vasek, J. Wadleigh and T. Moore, "Hacking Is Not Random: A Case-Control Study of Webserver-Compromise Risk," in IEEE Transactions on Dependable and Secure Computing, vol. 13, no. 2, pp. 206-219, 1 March-April 2016.

[26] M. Cui, J. Wang and B. Chen, "Flexible Machine Learning-Based Cyberattack Detection Using Spatiotemporal Patterns for Distribution Systems," in IEEE Transactions on Smart Grid, vol. 11, no. 2, pp. 1805-1808, March 2020.

[27] Y. Lu and L. D. Xu, "Internet of Things (IoT) Cybersecurity Research: A Review of Current Research Topics," in IEEE Internet of Things Journal, vol. 6, no. 2, pp. 2103-2115, April 2019.