✍️ 🅾🅲 Wordpress: Others Tolls For Enumeration

Intro

In the previous section, we saw how specialized tools like **WPScan** can be used to enumerate a ***WordPress*** website. In this section, we'll explore both techniques and automated tools that will help us enumerate ***WordPress*** sites in certain circumstances or, alternatively, complement the analysis performed with **WPScan** and thus gain a broader perspective before moving on to the exploitation phase.

Without further ado, let's look at these techniques/tools.

👁️ 🔢 Nmap Script Engine (NSE)

NSE scripts are scripts written in Lua that *extend Nmap's capabilities* beyond port scanning. They allow for enumeration, vulnerability detection, basic exploitation, and more. Nmap has scripts very useful to enumerate ***WordPress*** sites that are in the path **/usr/share/nmap/scripts/wordpress**. Those scripts are that are:

1) **http-wordpress-brute.nse:** Used to perform brute-force attacks against the ***WordPress*** login form (/wp-login.php). The goal is to find valid credentials (username + password) using a dictionary.
2) **http-wordpress-enum.nse:** Useful for listing installed versions of ***WordPress*** , **plugins**, and **themes** to identify vulnerable components and plan for potential attack vectors.
3) **http-wordpress-users.nse:** Useful for enumerating WordPress users using various techniques (such as author ID scanning, REST API leaks, etc.) to obtain a list of usernames, useful for brute force or other types of attacks

For example, to enumerated version of plugins and themes with **http-wordpress-enum.nse** we can use the command:

nmap -p80 --script http-wordpress-enum --script-args=http-wordpress-enum.root=/wordpress http://Target.WebSite/

Machine #1

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sV -p8585 --script http-wordpress-enum --script-args=http-wordpress-enum.root=/wordpress 192.168.171.128

[sudo] password for kali:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-12 12:01 EDT
Nmap scan report for 192.168.171.128
Host is up (0.00061s latency).

PORT     STATE SERVICE VERSION
8585/tcp open  http    Apache httpd 2.2.21 ((Win64) PHP/5.3.10 DAV/2)
|_http-server-header: Apache/2.2.21 (Win64) PHP/5.3.10 DAV/2
| http-wordpress-enum:
| Search limited to top 100 themes/plugins
|   plugins
|     akismet
|     ninja-forms 2.9.42
|   themes
|     twentyfourteen 1.8
|     twentyfifteen 1.6
|     twentysixteen 1.3
MAC Address: 00:0C:29:37:F0:B5 (VMware)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 24.77 seconds
```

Enumerated Plugins

Enumerated Themes

Machine #2

```
┌──(kali㉿kali)-[~]
└─$ nmap -p80 --script=http-wordpress-enum --script-args=http-wordpress-enum.root=/h3l105,search-limit=10000 symfonos.local

Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-13 18:51 EDT
Nmap scan report for symfonos.local (192.168.171.131)
Host is up (0.00057s latency).

PORT   STATE SERVICE
80/tcp open  http
| http-wordpress-enum:
| Search limited to top 4778 themes/plugins
|   plugins
|     akismet 4.1.2
MAC Address: 00:0C:29:05:2E:F0 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 5.62 seconds
```

Enumerated Plugins

As we can see we have successfully enumerated Some plugins and themes using this **Nmap NSE.** In the case of the **Machine #1** We have enumerated plugins and themes that we had not enumerated with **WPScan,** but in the case of **Machine #2**, the result we obtained was not better than what we obtained with **WPScan**, with which we were able to obtain its plugins and versions.

And we can do the same to enumerate users by just changing the script for **http-wordpress-users. In** the case of the machines we are using, this script will not work, but in other scenarios we can try to use it to see if we can get this information. The command that we can use is
nmap -sV -p80 --script http-wordpress-users
--script-args=http-wordpress-users.basepath=/wordpress/ target.website

Machine #1

```
Nmap done: 1 IP address (1 host up) scanned in 24.07 seconds

┌──(kali㉿kali)-[~]
└─$ sudo nmap -sV -p8585 --script http-wordpress-users --script-args=http-wordpress-users.basepath=/wordpress/ 192.168.171.128
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-16 10:45 EDT
Nmap scan report for 192.168.171.128
Host is up (0.00087s latency).

PORT     STATE SERVICE VERSION
8585/tcp open  http    Apache httpd 2.2.21 ((Win64) PHP/5.3.10 DAV/2)
| http-wordpress-users:
| Username found: admin
| Username found: vagrant
| Username found: user
| Username found: manager
|_Search stopped at ID #25. Increase the upper limit if necessary with 'http-wordpress-users.limit'
|_http-server-header: Apache/2.2.21 (Win64) PHP/5.3.10 DAV/2
MAC Address: 00:0C:29:37:F0:B5 (VMware)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 45.85 seconds
```

Users Enumerated

In this case, we have successfully enumerated users by using this **NSE**

Notes📝:

1) A important note to mention is that the ***WordPress*** isn't accessible by the root domain name but via **/wordpress** o any other apart kind **/** , we can use the argument **--script-args=http-wordpress-enum.root=/wordpress** to enumerate plugins, and **--script-args=http-wordpress-users.basepath=/wordpress** to enumerate users, , as we saw in the examples of **Machine #1** & **Michanie #2**. If ***WordPress*** is accessible by the root domain we don't need to user this argument

2) Another thing to mention is that **Nmap** will use a list of the top 100 plugins and themes by default. Adding the command **--script-args=** the argument ***search-limit=*** we can indicate how aggressive we want the analysis to be, having, at the moment that we write this article, a limit of **4778.** like **--script-args=search-limit=**

🗄️🔫 Directory Listing

We can use this technique ***Directory Listing*** to enumerate plugins if we have access to a part **/wp-content/plugins/** using tools like **Dirb**, **gobuster**, **feroxbuster** or **Dirbuster** if we want to user a graphical interface tool, passing a dictionary with a list of ***WordPress*** plugins.

In this case we will use **gobuster** and the dictionary **wp-plugins** of 🦬 **metasploit** with the command:

gobuster dir -u http://target.website/wp-content/plugins/ -w /usr/share/wordlists/metasploit/wp-plugins.txt

Machine #1

```
┌──(kali㉿kali)-[~]
└─$ gobuster dir -u http://192.168.171.128:8585/wordpress/wp-content/plugins/ -w /usr/share/wordlists/metasploit/wp-plugins.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.171.128:8585/wordpress/wp-content/plugins/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/metasploit/wp-plugins.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/akismet              (Status: 403) [Size: 238]
/aux                  (Status: 403) [Size: 234]
/ninja-forms          (Status: 301) [Size: 277] [--> http://192.168.171.128:8585/wordpress/wp-content/plugins/ninja-forms/]
Progress: 104063 / 104064 (100.00%)
===============================================================
Finished
===============================================================
```

Enumerated Plugins

Machine #2

```
┌──(kali㉿kali)-[~]
└─$ gobuster dir -u http://symfonos.local/h3l105/wp-content/plugins/ -w /usr/share/wordlists/metasploit/wp-plugins.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://symfonos.local/h3l105/wp-content/plugins/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/metasploit/wp-plugins.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/akismet              (Status: 301) [Size: 344] [--> http://symfonos.local/h3l105/wp-content/plugins/akismet/]
/mail-masta           (Status: 301) [Size: 347] [--> http://symfonos.local/h3l105/wp-content/plugins/mail-masta/]
/site-editor          (Status: 301) [Size: 348] [--> http://symfonos.local/h3l105/wp-content/plugins/site-editor/]
Progress: 104063 / 104064 (100.00%)
===============================================================
Finished
===============================================================
```

Enumerated Plugins

And we have enumerated plugins for our machine. But we can go further.

Once we have enumerated some plugins via directory listing we can open the link in the browser and adding **/readme.txt** we can see the version of that plugin:

Machine #1

```
192.168.171.128:8585/wordpress/wp-content/plugins/ninja-forms/readme.txt
Kali Linux  Kali Tools  Kali Docs  Kali Forums  Kali NetHunter  Exploit-DB  Google Hacking DB  OffSec

=== Ninja Forms ===
Contributors: wpninjasllc, kstover, jameslaws, wpnzach, kbjohnson90, aman086, daveshine, mordauk, bftrick, helgatheviking
Tags: form, forms, contact form, custom form, form builder, form creator, form manager, form creation, contact forms, custom forms, forms builder, forms
Requires at least: 4.3
Tested up to: 4.5
Stable tag: 2.9.42
License: GPLv2 or later
```

Plugin Version

Machine #2

```
symfonos.local/h3l105/wp-content/plugins/site-editor/readme.txt
Kali Linux  Kali Tools  Kali Docs  Kali Forums  Kali NetHunter  Exploit-DB  Google Hacking DB  OffSec

=== Site Editor - WordPress Site Builder - Theme Builder and Page Builder ===
Contributors: wpsiteeditor
Tags: site editor, site builder, page builder, theme builder, theme framework, design, inline editor, inline text editor, layout builder,live options, live, customizer, theme customizer, header builder, f
editor, options framework, front end, page builder plugin, builder, responsive, front end editor, landing page, editor, drag-and-drop, shortcode, wordpress, ultra flexible, unlimited tools, elements, modu
theme options, video backgrounds, font awesome, Optimized, fast, quick, ux, ui
Requires at least: 4.7
Tested up to: 4.7.4
Stable tag: 1.1.1
License: GPLv3
License URI: https://www.gnu.org/licenses/gpl-3.0.html
```

Plugin Version

Now it only remains to investigate whether these versions have any vulnerabilities.

🧑‍🌐 WhatWeb

WhatWeb is a tool to identify technologies of a web site. In the context of a wordpress ***WordPress*** website we will use it to identify the ***WordPress*** version that is running in that website. We can do it with the command:

whatweb http://targetwebsite.com/

Machine #1

```
┌──(kali㉿kali)-[~]
└─$ whatweb http://192.168.171.128:8585/wordpress/
http://192.168.171.128:8585/wordpress/ [200 OK] Apache[2.2.21], Cookies[nf_wp_session], Country[RESERVED][ZZ], HTML5, HTTPServer[Apache/2.2.21 (W
in64) PHP/5.3.10 DAV/2], IP[192.168.171.128], JQuery[1.12.4], MetaGenerator[WordPress 4.6.1], PHP[5.3.10], PoweredBy[WordPress,WordPress,], Scrip
t[text/javascript], Title[Metasploit3 | Oh hai, is this metasploitable?], UncommonHeaders[link], WebDAV[2], WordPress[4.6.1], X-Powered-By[PHP/5.
3.10]
```

Machine 2

```
┌──(kali㉿kali)-[~]
└─$ whatweb http://symfonos.local/h3l105/
http://symfonos.local/h3l105/ [200 OK] Apache[2.4.25], Country[RESERVED][ZZ], HTML5, HTTPServer[Debian Linux][Apache/2.4.25 (Debian)], IP[192.168
.171.131], JQuery, MetaGenerator[WordPress 5.2.2], PoweredBy[WordPress,WordPress,], Script[text/javascript], Title[helios site &#8211; Just anoth
er WordPress site], UncommonHeaders[link], WordPress[5.2.2]
```

Conclusions

As we have seen, there are several ways to enumerate a ***WordPress*** site that will serve to complement the analysis done by **WPScan**.

And with this section, we conclude the recognition phase and move on to the most critical phase, the **exploitation phase.**