



Intro

Today, the most important tool for scanning **WordPress** instances and performing security tests on web applications that implement this technology is **WPScan**.

WPScan is a tool that allows us to *fully automate* all the processes we saw in the manual enumeration section. We can enumerate **users**, **directories**, as well as the **WordPress version** running on a web application, **plugins** and **themes** and their versions—all of this, I repeat, *completely automatically*.

We can also perform a brute-force attack on an exposed WordPress login to gain access to said panel, something we'll cover in another section.

Also, with this tool, we can enumerate system sections, among other things.

This tool comes installed by default 🐧 in Kali and will be used from the terminal.

Without further ado, let's learn how to use WPScan.

WPScan in action

To use a generic scan using **WPScan** we just have to run the next command

wpscan --url http://Target.WebSite/

With the result:

Machine #1

```
Interesting Findings()
[+] Headers
  Interesting Entries:
    - Server: Apache/2.2.22 (Ubuntu) PHP/5.3.10-0ubuntu2
    - X-Powered-By: PHP/5.3.10
  Found By: Headers (Passive Detection)
  Confidence: 100%

[+] XML-RPC seems to be enabled: http://192.168.171.128:8080/wordpress/xmlrpc.php
  Found By: Link Tag (Passive Detection)
  Confidence: 100%
  Confirmed By: Direct Access (Aggressive Detection), 100% confidence
  References:
    - http://codex.wordpress.org/XML-RPC_rpc_methods
    - 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_pingback_access/
    - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access/

[+] WordPress readme found: http://192.168.171.128:8080/wordpress/readme.html
  Found By: Direct Access (Aggressive Detection)
  Confidence: 100%

[+] Full Path Disclosure found: http://192.168.171.128:8080/wordpress/wp-includes/rss-functions.php
  Interesting Entry: C:\wamp\www\wordpress/wp-includes/rss-functions.php
  Found By: Direct Access (Aggressive Detection)
  Confidence: 100%
  Reference: https://www.cwasp.org/index.php/Full_Path_Disclosure

[+] Upload directory has listing enabled: http://192.168.171.128:8080/wordpress/wp-content/uploads/
  Found By: Direct Access (Aggressive Detection)
  Confidence: 100%

[+] The external WP-Cron seems to be enabled: http://192.168.171.128:8080/wordpress/wp-cron.php
  Found By: Direct Access (Aggressive Detection)
  Confidence: 0%
  References:
    - https://www.kalilinux.net/defend-wordpress-from-ddos
    - https://github.com/wpscanteam/wpscan/issues/129

[+] WordPress version 4.9.1 identified (Insecure, released on 2016-09-07)
  Found By: Rss Generator (Passive Detection)
  - http://192.168.171.128:8080/wordpress/index.php/feed/, generator=http://wordpress.org/4.9.1/generator/
  - http://192.168.171.128:8080/wordpress/index.php/comment/feed/, generator=http://wordpress.org/4.9.1/generator/
  - http://192.168.171.128:8080/wordpress/index.php/comment/feed/, generator=http://wordpress.org/4.9.1/generator/

[+] WordPress theme in use: twentyfourteen
  Location: http://192.168.171.128:8080/wordpress/wp-content/themes/twentyfourteen/

Technology versions
WordPress Version
WordPress Theme
```


Machine #2

```
[+] Plugin(s) Identified:

[+] mail-masta
  Location: http://symfonos.local/h3185/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 (80% confidence)
  Found By: Readme - Stable Tag (Aggressive Detection)
  - http://symfonos.local/h3185/wp-content/plugins/mail-masta/readme.txt

[+] site-editor
  Location: http://symfonos.local/h3185/wp-content/plugins/site-editor/
  Latest Version: 1.1.1 (up to date)
  Last Updated: 2017-05-02T23:36:00.000Z
  Found By: Urls In Homepage (Passive Detection)
  Version: 1.1.1 (80% confidence)
  Found By: Readme - Stable Tag (Aggressive Detection)
  - http://symfonos.local/h3185/wp-content/plugins/site-editor/readme.txt

Plugins Enumerated
```

As we can see, we have enumerated a lot of interesting information like the [WordPress](#) version, directory listing, and even **plugins** and **theme** versions. With this information, we can investigate if those versions have **some of those software have a public exploit**, maybe *on internet* or using tools like **Searchsploit** (see the article  **Searchsploit**), and **go to the phase of exploitation**. All this information enumerated *completely automatically*.

It's important to mention that we used two different machines in this case. We weren't able to list any plugins on the first one, but we'll see other alternatives in other sections, both in terms of techniques and tools, to obtain this information.

But with WPScan we can go beyond. If we add the flag **-e** we can enumerate all that information and also **the users of that WordPress** with the command:

wpscan --url http://TargetWebSite/ -e

Manichine #1

```
[*] User(s) Identified:

[*] admin
| Found By: Author Posts - Author Pattern (Passive Detection)
| Confirmed By:
| Rss Generator (Passive Detection)
| Author Id Brute Forcing - Author Pattern (Aggressive Detection)
| Login Error Messages (Aggressive Detection)

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

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

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

And look, we have enumerated all the **users** that exist on the [WordPress](#) instance, *completely automatically*.

Conclusions

And there are some things we can do automatically with **WPscan**. We can do *more* things with this powerful tool, but in this section I want us to focus on **enumeration** and how we can do it completely automated. In future sections, we'll look at how we can do other cool things like **brute-force attacks**.