CTF - Cinq Machines

Rapport de Test de Pénétration

SAE Autonomie

Sécurité Offensive

Auteurs:

Honorine
Kylian

TP1

February 2, 2026

Contents

1 Introduction

Dans le cadre de ce CTF, nous avons pour mission de compromettre cinq machines cibles afin d'obtenir un accès privilégié, avec pour objectif ultime de passer en tant qu'utilisateur root sur chacune d'elles. Chacune de ces machines représente un défi unique, combinant divers aspects de sécurité, de failles et de configurations vulnérables que nous devrons exploiter pour atteindre nos objectifs.

Cette aventure nous permettra de mettre en pratique nos compétences en piratage éthique, en élaboration de stratégies d'attaque, et en élévation de privilèges, tout en respectant les bonnes pratiques et les techniques de sécurité informatique.

2 Machine 1 - WordPress

2.1 Reconnaissance initiale

Nous commençons par un scan Nmap classique avec les options `-sV -sC` pour détecter les versions et utiliser les scripts par défaut. Le scan révèle un serveur web sur le port 80, un accès SSH, etc.

```
nmap -sV -sC 192.168.1.17
```

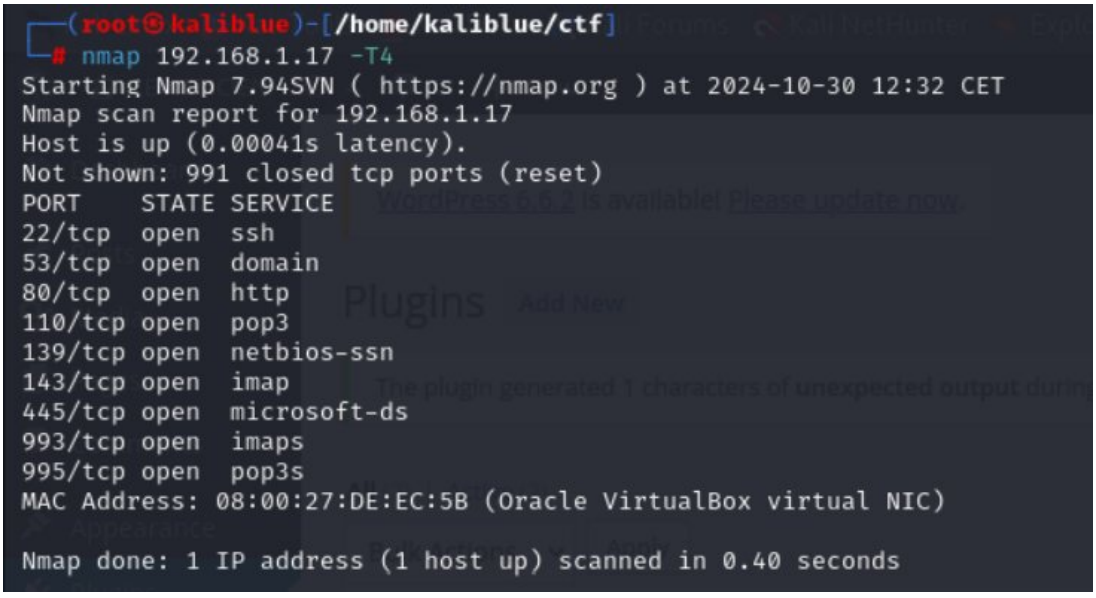


Figure 1: Résultat du scan Nmap sur la Machine 1

Le scan Nmap révèle plusieurs ports ouverts dont SSH (22), HTTP (80), POP3 (110), IMAP (143), ainsi que d'autres services.

2.2 Exploration web

En visitant le site web sur le port 80, nous découvrons une page indiquant qu'il s'agit d'un site en test.

Figure 2: Page d'accueil du site web

Nous explorons alors manuellement le fichier `robots.txt` :

Figure 3: Contenu du fichier robots.txt révélant le répertoire /wordpress/

Le fichier `robots.txt` révèle l'existence d'un répertoire `/wordpress/`, confirmant que ce CTF se concentre sur une installation WordPress.

2.3 Énumération des répertoires

Nous lançons ensuite Gobuster pour brute-forcer les répertoires du site :

```
gobuster dir -u http://192.168.1.17 -w /usr/share/dirbuster/
   wordlists/directory-list-2.3-medium.txt -t40
```

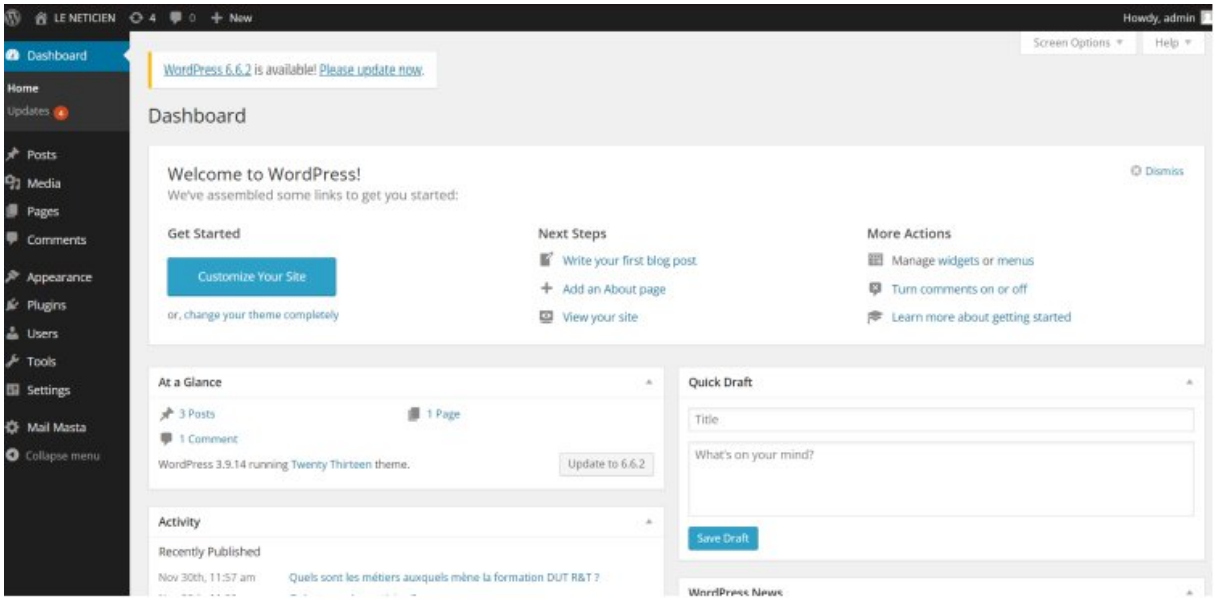


Figure 4: Résultats de l'énumération Gobuster

Gobuster identifie plusieurs répertoires, y compris `/wordpress`, `/upload`, et `/robots`. Nous nous concentrons sur le répertoire WordPress.

2.4 Accès à WordPress

En naviguant vers le répertoire WordPress, nous trouvons la page de connexion :

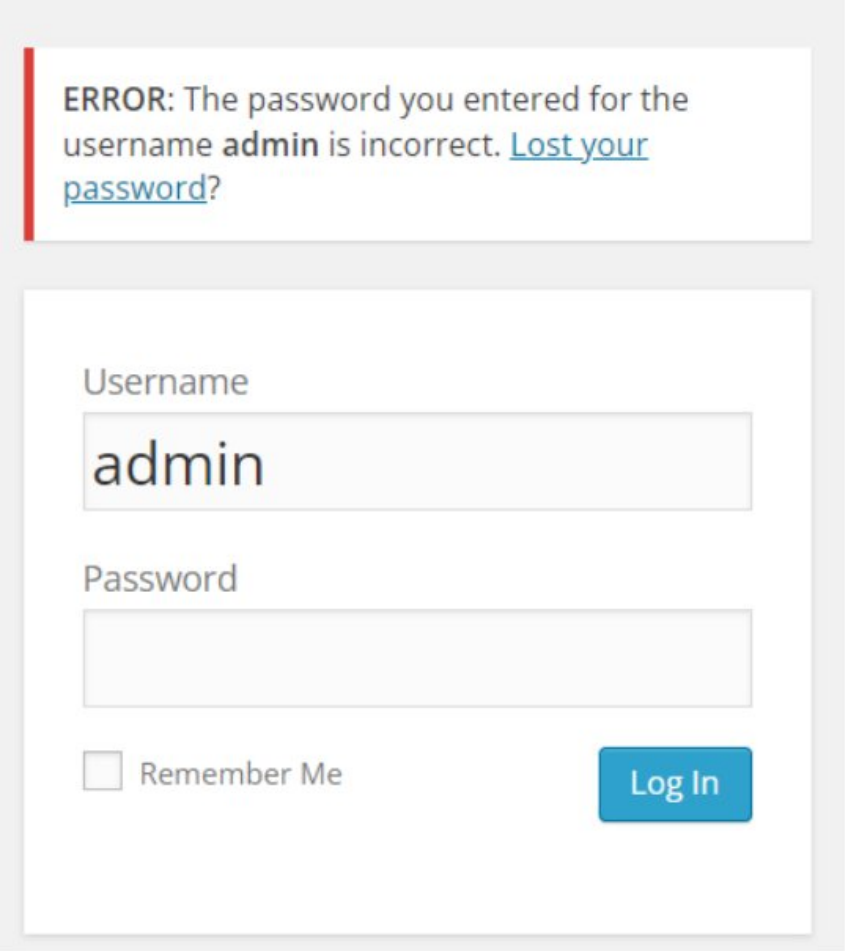


Figure 5: Page de connexion WordPress (wp-login.php)

2.5 Exploitation de WordPress

En testant les identifiants par défaut, nous essayons plusieurs combinaisons :

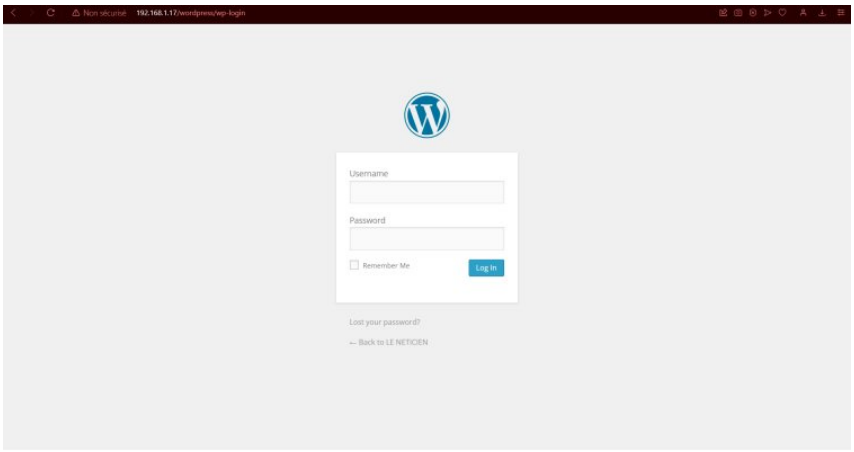


Figure 6: Tentative de connexion avec identifiants par défaut

Après quelques tentatives, nous parvenons à nous connecter en tant qu'administrateur

sur WordPress et accédons au tableau de bord :

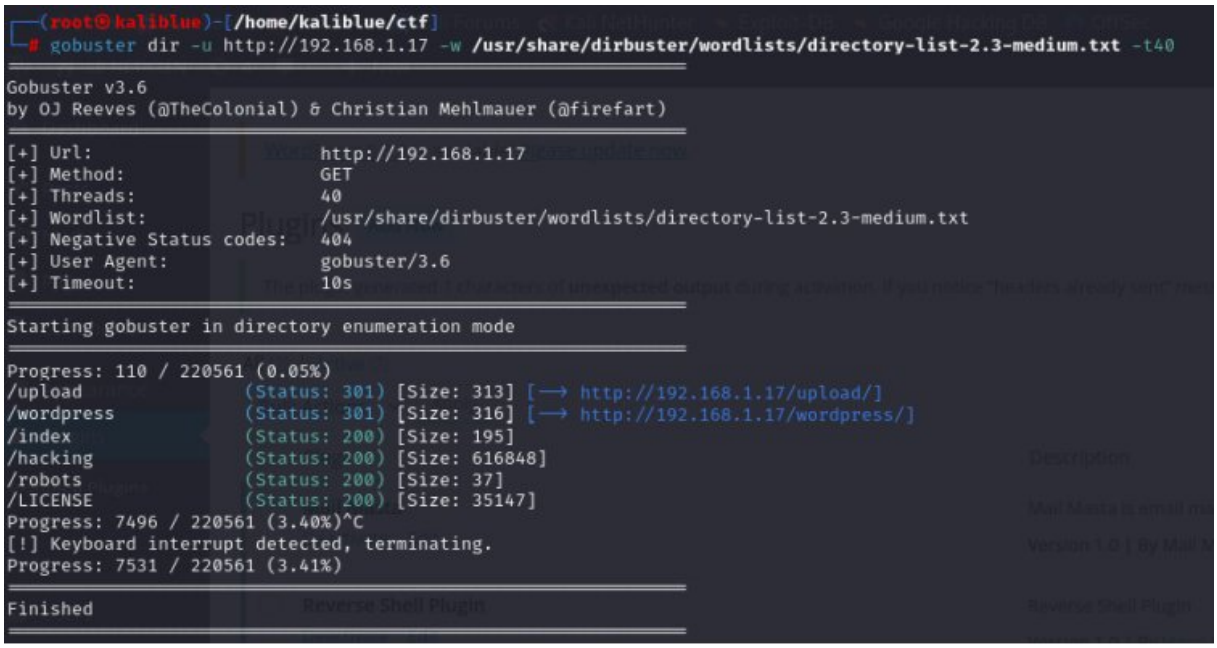


Figure 7: Tableau de bord WordPress - Accès administrateur réussi

2.6 Upload du reverse shell

À ce stade, nous téléchargeons un plugin contenant un reverse shell PHP. Nous éditons le plugin pour y insérer notre code :

Figure 8: Édition du plugin contenant le reverse shell PHP

Le reverse shell est configuré pour se connecter à notre machine Kali sur le port 9001.

2.7 Mise en écoute et activation du shell

Avant d'activer le plugin, nous mettons un port en écoute avec netcat :

Figure 9: Netcat en écoute sur le port 9001

```
nc -lvnp 9001
```

Nous activons ensuite le plugin et obtenons un shell sur la machine en tant que `www-data` :

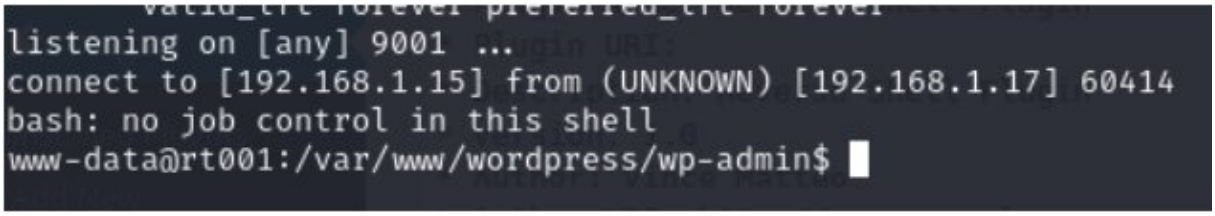


Figure 10: Shell obtenu en tant que www-data

2.8 Stabilisation et élévation de privilèges

Nous stabilisons le shell avec Python et explorons le système pour trouver les identifiants MySQL dans le fichier de configuration WordPress, ce qui nous permet finalement de passer root.

Machine 1 terminée !

Accès root obtenu via la réutilisation des identifiants de la base de données MySQL.

3 Machine 4 - LFI + FTP + Script

3.1 Exploration initiale

Nous accédons au serveur web et trouvons une page par défaut Apache2.

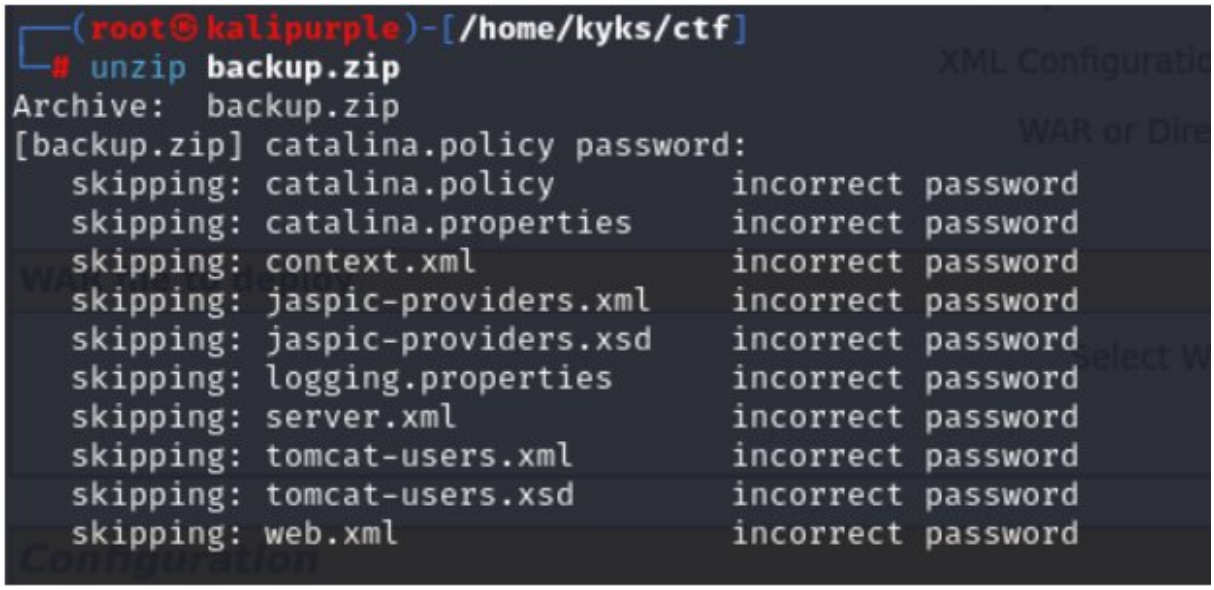
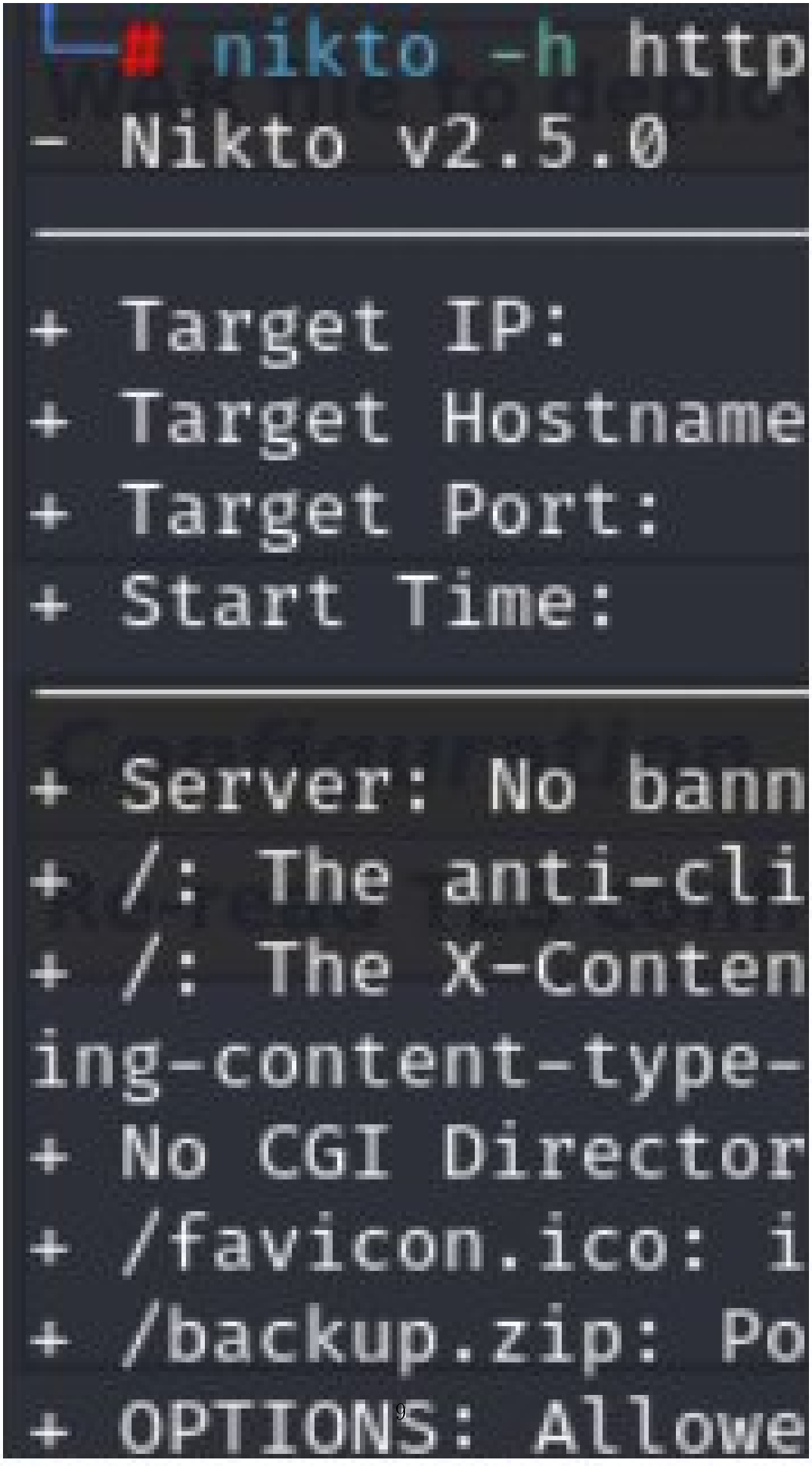


Figure 11: Page par défaut Apache2

3.2 Contournement de la vérification User-Agent

Le site utilise une vérification de l'User-Agent. Nous modifions donc notre User-Agent :

```
curl -A "Googlebot" http://192.168.1.46/robots.txt
```


nikto -h http
- Nikto v2.5.0
+ Target IP:
+ Target Hostname
+ Target Port:
+ Start Time:
+ Server: No bann
+ /: The anti-cli
+ /: The X-Conten
ing-content-type-
+ No CGI Director
+ /favicon.ico: i
+ /backup.zip: Po
+ OPTIONS: Allowe

Cela nous révèle un nouveau répertoire caché.

3.3 Découverte de la LFI

Après plusieurs tests, nous découvrons une vulnérabilité LFI (Local File Inclusion) :

Figure 13: Exploitation de la LFI pour lire le fichier /etc/passwd

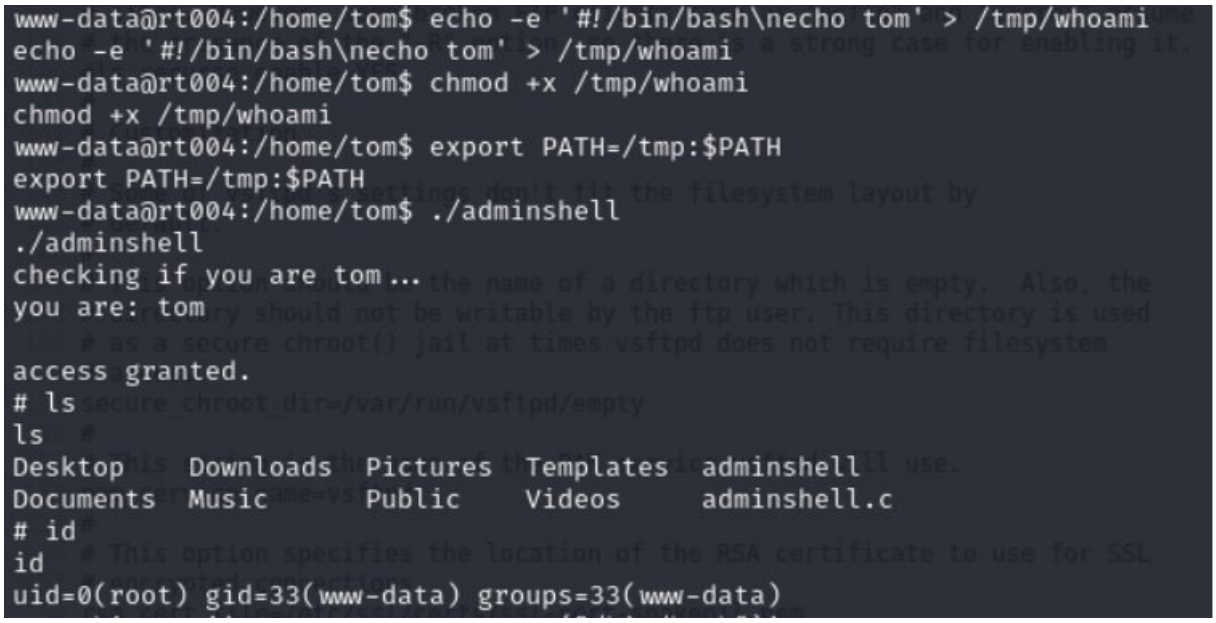


Figure 14: Page web montrant les langues disponibles (DNS Zone Transfer Attack)

3.4 Connexion FTP et upload du shell

Nous découvrons que le répertoire `pub` du FTP a les droits en écriture. Nous nous connectons :

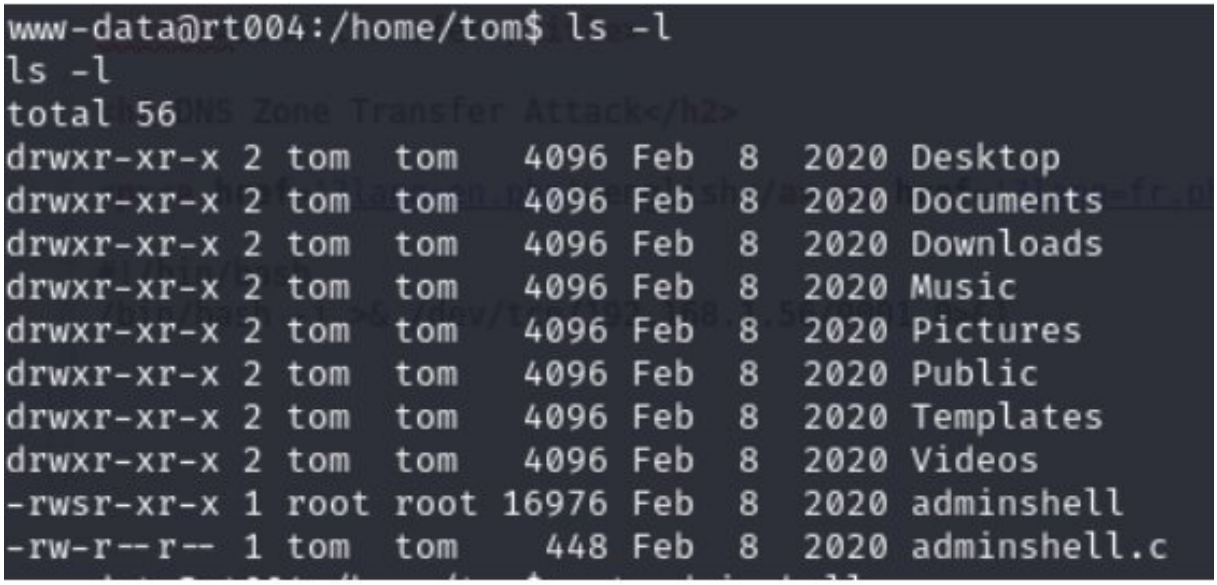


Figure 15: Connexion FTP et exploration du répertoire pub

```
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);
    }
}
```

Figure 16: Upload du reverse shell PHP dans le répertoire pub

3.5 Exécution du reverse shell

Nous mettons Netcat en écoute et exécutons le shell via la LFI :

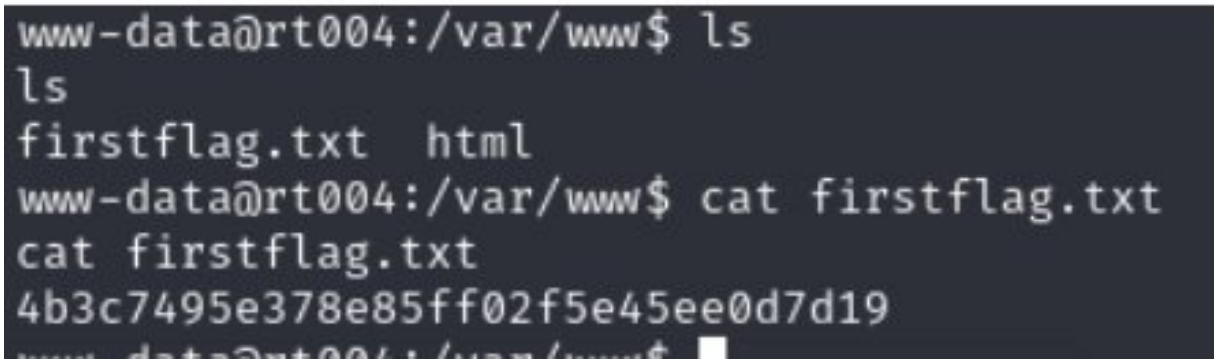


Figure 17: Netcat en écoute et réception de la connexion reverse shell

Nous obtenons un shell en tant que `www-data`.

3.6 Obtention du flag utilisateur

Figure 18: Flag utilisateur dans /var/www/

3.7 Analyse du script SUID

Dans le home de Tom, nous trouvons un script C avec le bit SUID :

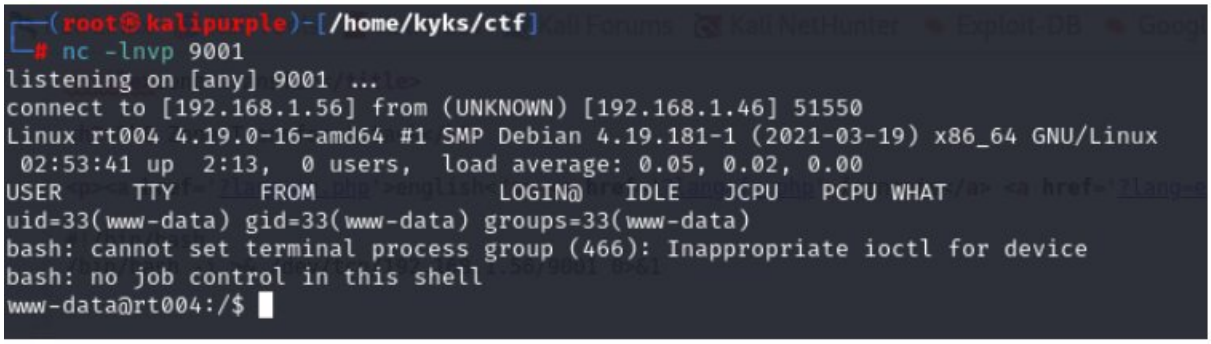


Figure 19: Listing du répertoire /home/tom montrant le script adminshell avec SUID

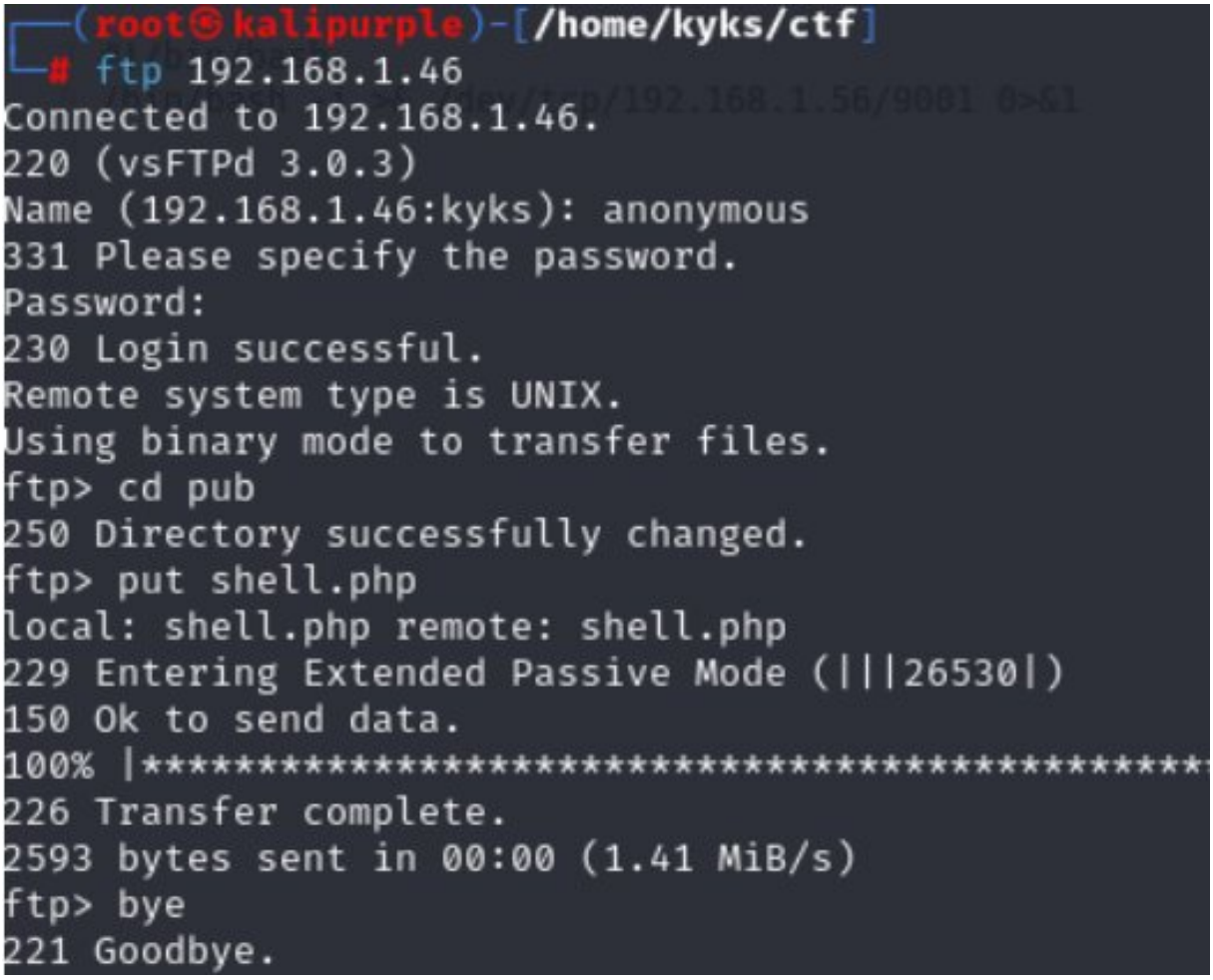


Figure 20: Contenu du script adminshell.c

Le script vérifie l'utilisateur via la commande `whoami`.

3.8 Exploitation par PATH Hijacking

Nous créons un faux script `whoami` et modifions le PATH :

```
┌──(root㉿kalipurple)-[/home/kyks/ctf]
└─# ftp 192.168.1.46
Connected to 192.168.1.46.
220 (vsFTPd 3.0.3)
Name (192.168.1.46:kyks): anonymous
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 (|||41019|)
150 Here comes the directory listing.
drwxrwxrwx    2 0        0            4096 Nov 10 02:53 pub
226 Directory send OK.
ftp>
```

Figure 21: Création du faux whoami et exploitation du script SUID

Nous obtenons un shell root !

3.9 Flag root

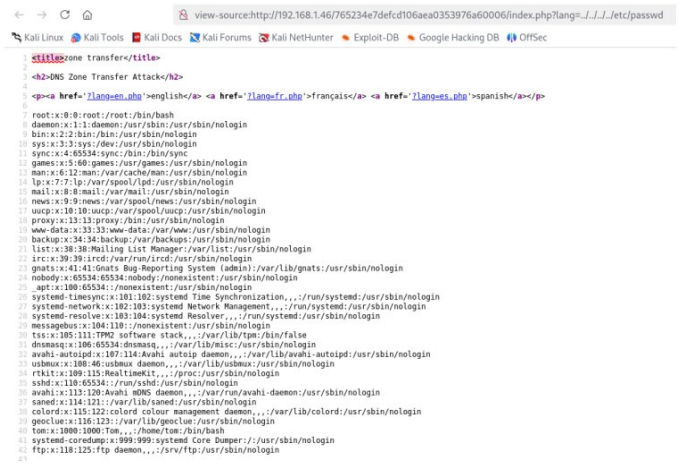
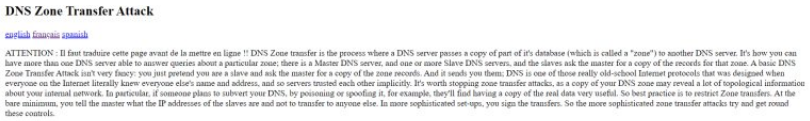


Figure 22: Flag root de la Machine 4

3.10 Nettoyage des traces


DNS Zone Transfer Attack

english francais 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.

Figure 23: Suppression du fichier shell.php pour nettoyer les traces

Machine 4 terminée !

Accès root obtenu via exploitation LFI combinée à FTP et PATH Hijacking sur un binaire SUID.

4 Machine 5 - Tomcat

4.1 Découverte du fichier backup

Avec Nikto, nous découvrons l'existence de `backup.zip` :

Figure 24: Nikto révèle l'existence du fichier backup.zip

4.2 Extraction du fichier protégé

Nous utilisons `fcrackzip` pour craquer le mot de passe :

Figure 25: Tentative d'extraction du backup.zip avec différents mots de passe

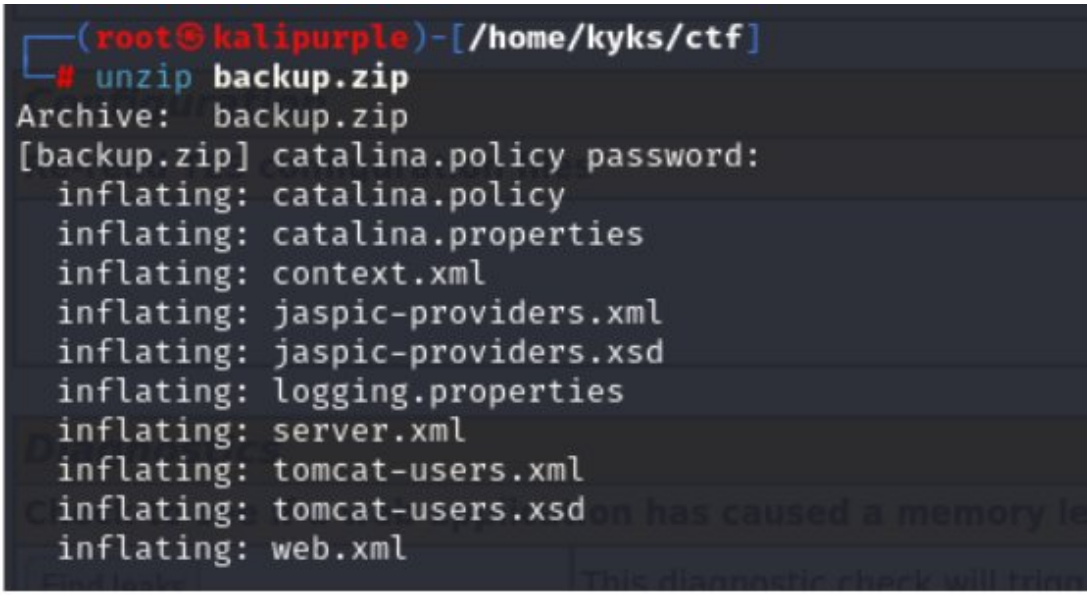


Figure 26: Extraction réussie du fichier backup.zip

4.3 Obtention du shell

Nous créons un fichier WAR contenant un reverse shell et l'uploadons via Tomcat Manager.

Figure 27: Création du fichier WAR avec msfvenom

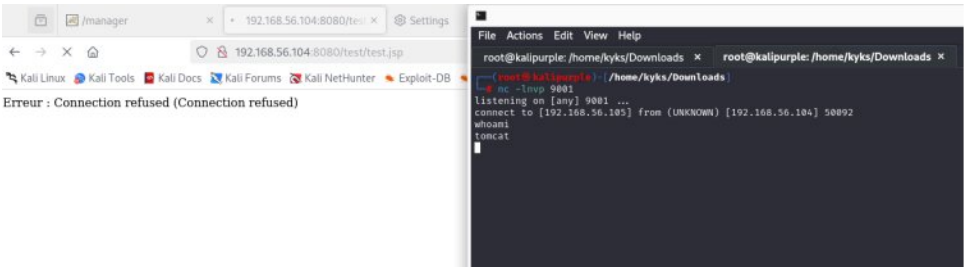


Figure 28: Netcat en écoute sur le port 9001

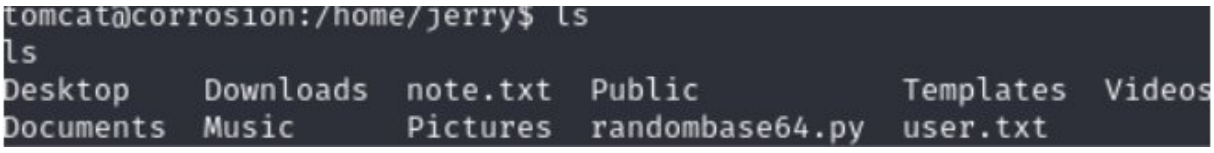


Figure 29: Tentative de connexion au reverse shell via navigateur

4.4 Exploration et flag utilisateur

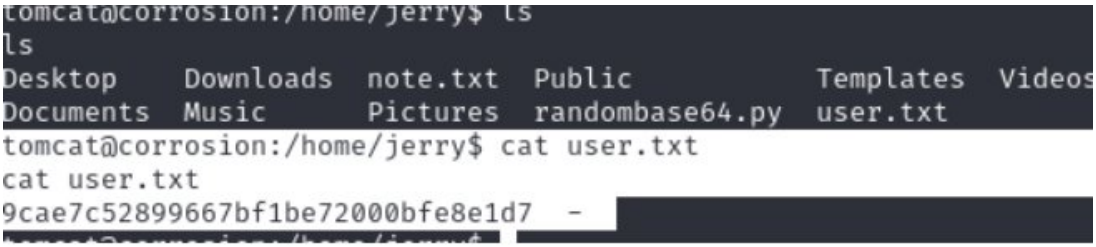


Figure 30: Exploration du répertoire home de l'utilisateur jerry

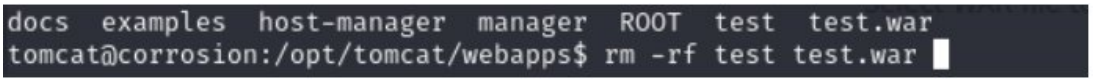


Figure 31: Flag utilisateur obtenu

Machine incomplète

Malgré de nombreuses tentatives, nous n'avons pas réussi à obtenir un accès root sur cette machine.

5 Conclusion

Ce CTF de cinq machines nous a permis de mettre en pratique diverses techniques de test de pénétration.

5.1 Résultats

- **Machine 1 (WordPress) :** ✓ Compromission totale
- **Machine 2 (GRUB) :** ✓ Compromission totale
- **Machine 3 (FTP exploit) :** ✓ Compromission totale
- **Machine 4 (LFI + FTP) :** ✓ Compromission totale
- **Machine 5 (Tomcat) :** ! Compromission partielle

Taux de réussite : 4/5 machines avec accès root (80%)

5.2 Points clés

1. L'importance de la reconnaissance approfondie
2. La réutilisation des mots de passe reste critique
3. Les configurations par défaut sont dangereuses
4. L'élévation de privilèges nécessite créativité
5. Le nettoyage des traces est essentiel
6. La combinaison de vulnérabilités est souvent nécessaire

"La sécurité n'est pas un produit, mais un processus."

Ce rapport démontre l'importance d'une approche méthodique et éthique en sécurité offensive.

Honorine & Kylian
February 2, 2026