

# **Rapport de projet**

## **Durcissement (Hardening) et Sécurisation d'une Infrastructure Serveur Linux**

**Réalisé par : MABENGO Gloire Précieux**

## SOMMAIRE

I.	Présentation du projet .....	2
II.	Implémentation .....	2
1.	Mise en place du réseau et des services .....	2
1.1.	Configuration réseau .....	2
1.2.	Déploiement des services de base .....	3
2.	Scan de port .....	3
3.	Changement et blocages de port par défaut (SSH et HTTP) .....	3
3.1.	Sécurisation du SSH (Port 2222 et UFW) .....	3
3.2.	Sécurisation Web (HTTPS et Redirection) .....	4
3.3.	Défense Active (Fail2Ban) .....	4
III.	Conclusion .....	5
IV.	Webographie .....	5

## I. Présentation du projet

Ce projet a pour objectif la sécurisation d'un serveur Ubuntu hébergeant un service Web Apache et un accès distant SSH. L'enjeu est de passer d'une configuration par défaut, vulnérable aux scans et aux interceptions, à une infrastructure robuste.

## II. Implémentation

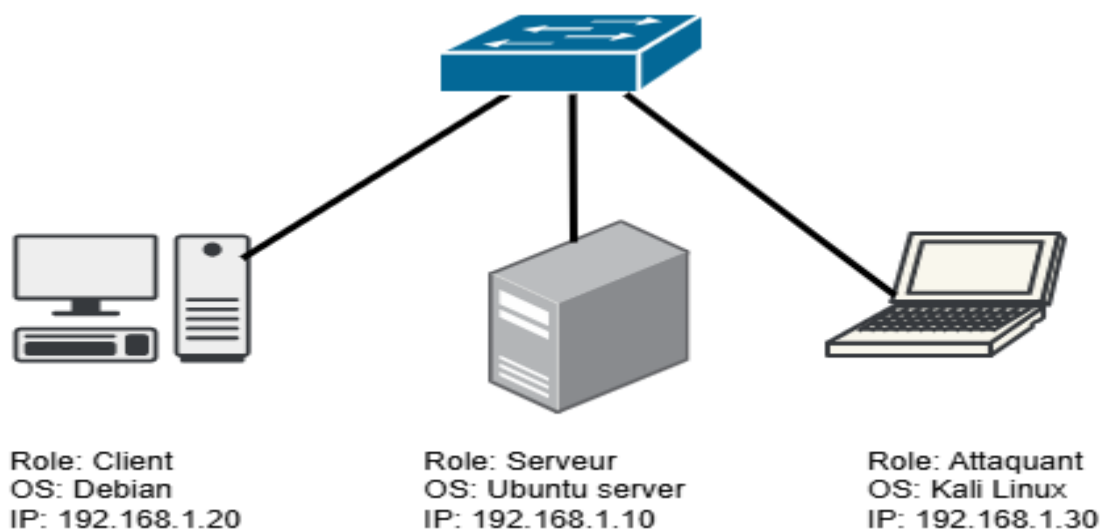
Le projet s'appuie sur une architecture virtualisée (VMWARE Workstation 25H2) comprenant trois entités :

- **Le Serveur (Cible)** : Ubuntu Server, configuré avec une IP statique.
- **Le Client** : Debian, seule machine autorisée à administrer le serveur.
- **L'Attaquant (Auditeur)** : Kali Linux, utilisé pour tester la visibilité des services et simuler des attaques.

### 1. Mise en place du réseau et des services

#### 1.1. Configuration réseau

- **Mise en place d'un LAN dans VMWARE avec l'option Lan Segment (Lab-Sec1)**



- Fixation de l'adresse IP sur le serveur (modification du fichier `/etc/netplan/00-installer-config.yaml`)

```
# This is the network config written by 'subiquity'
network:
  version: 2
  ethernet:
    ens33:
      dhcp4: false
      addresses:
        - 192.168.1.10/24
```

## 1.2. Déploiement des services de base

- Installation d'Apache2 pour le service Web
- Installation d'OpenSSH pour l'administration à distance

```
prcx23@ubuntu-server: ~
prcx23@debian:~$ ssh prcx23@192.168.1.10
prcx23@192.168.1.10's password:
Welcome to Ubuntu 25.10 (GNU/Linux 6.17.0-12-generic x86_64)

* Documentation:  https://docs.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/pro

System information as of Sun Feb 15 13:29:44 UTC 2026

System load: 0.0           Memory usage: 38%    Processes:      228
Usage of /:  45.5% of 9.75GB Swap usage:   0%      Users logged in: 0

70 mises à jour peuvent être appliquées immédiatement.
15 de ces mises à jour sont des mises à jour de sécurité.
Pour afficher ces mises à jour supplémentaires, exécuter : apt list --upgradable

Last login: Sun Feb 15 13:23:42 2026 from 192.168.1.20
prcx23@ubuntu-server:~$
```

## 2. Scan de port

Un premier scan Nmap depuis Kali Linux a révélé la visibilité totale des services :

- **Port 80 (HTTP) :** Ouvert (trafic en clair).
- **Port 22 (SSH) :** Ouvert (cible de brute-force).

- **Constat** : L'attaquant peut identifier les versions des services et tenter des interceptions de mots de passe sur le flux HTTP.

```

kali@kali: ~
Session Actions Edit View Help

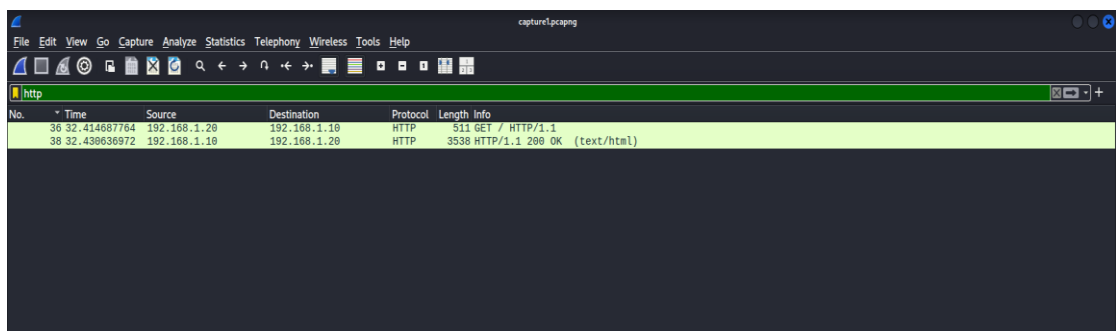
(kali@kali)-[~]
$ nmap -sV 192.168.1.10
Starting Nmap 7.95 ( https://nmap.org ) at 2026-02-15 08:10 EST
Nmap scan report for 192.168.1.10
Host is up (0.0010s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 10.0p2 Ubuntu 5ubuntu5 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.64 ((Ubuntu))
MAC Address: 00:0C:29:F2:A9:2F (VMware)
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 24.52 seconds

(kali@kali)-[~]
$ 

```

- Vue du trafic HTTP avec Wireshark



### 3. Changement et blocages de port par défaut (SSH et HTTP)

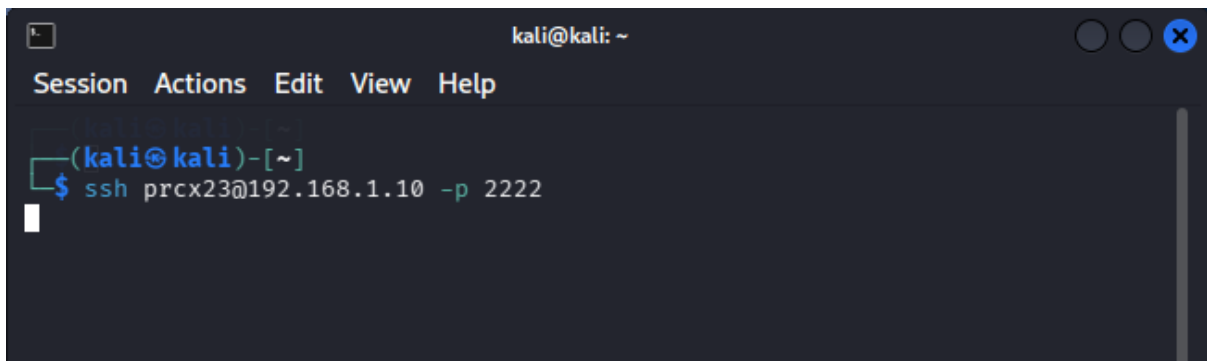
#### 3.1. Sécurisation du SSH (Port 2222 et UFW)

- **Démarche** : Déplacement du port SSH de 22 vers **2222** pour éviter les scripts automatisés. (Remplacer la ligne Port 22 en Port 2222 dans le fichier `/etc/ssh/sshd_config` )

```
# When systemd socket activation is used (the default), the socket
# configuration must be re-generated after changing Port, AddressFamily, or
# ListenAddress.
#
# For changes to take effect, run:
#
#   systemctl daemon-reload
#   systemctl restart ssh.socket
#
Port 2222
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key
```

- **Blocage par Pare-feu (UFW) :** Fermeture du port 2222 pour tout le réseau, sauf pour l'IP de la machine Debian (192.168.1.20).



```
kali@kali: ~
Session Actions Edit View Help
(kali@kali)-[~]
$ ssh prcx23@192.168.1.10 -p 2222
```

Constat : la machine attaquante n'arrive pas à accéder au service ssh

### 3.2. Sécurisation Web (HTTPS et Redirection)

- **SSL/TLS :** Génération d'un certificat auto-signé pour activer le HTTPS (Port 443).

```
#!/bin/bash

KEY_PATH="/etc/ssl/private/apache.key"
CERT_PATH="/etc/ssl/certs/apache.crt"

echo "### Génération du certificat ....###"
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
    -keyout $KEY_PATH \
    -out $CERT_PATH

echo "Génération terminée"
~
~
```

- **Redirection 301** : Configuration d'Apache pour rediriger automatiquement tout visiteur du port 80 vers le port 443.

Nous avons utilisé le module rewrite d'Apache :

- **Fichier de configuration** : /etc/apache2/sites-available/000-default.conf

```
#Rediriger le http vers https
RewriteEngine On
RewriteCond %{HTTPS} off
RewriteRule ^(.*)$ https://%{HTTP_HOST}%{REQUEST_URI} [L,R=301]
</VirtualHost>
~
~
~
~
~
```

**Directives ajoutées :**

- **Problème rencontré** : Wireshark affichait encore du trafic HTTP.
- **Analyse** : Le trafic HTTP persistait car il contenait la réponse "301 Moved Permanently". Les données sensibles, elles, étaient bien transférées dans le tunnel TLS (HTTPS).

993 864.818703755 192.168.1.20	192.168.1.10	HTTP	419 GET / HTTP/1.1
995 864.819891348 192.168.1.10	192.168.1.20	HTTP	631 HTTP/1.1 301 Moved Permanently (text/html)

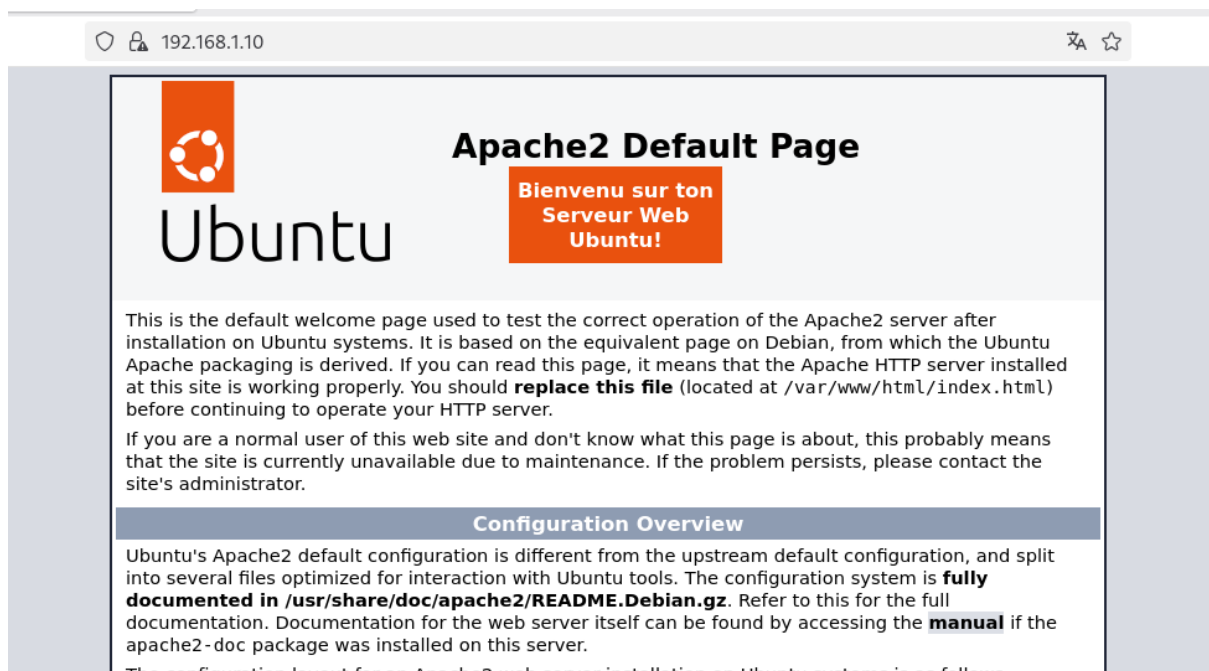
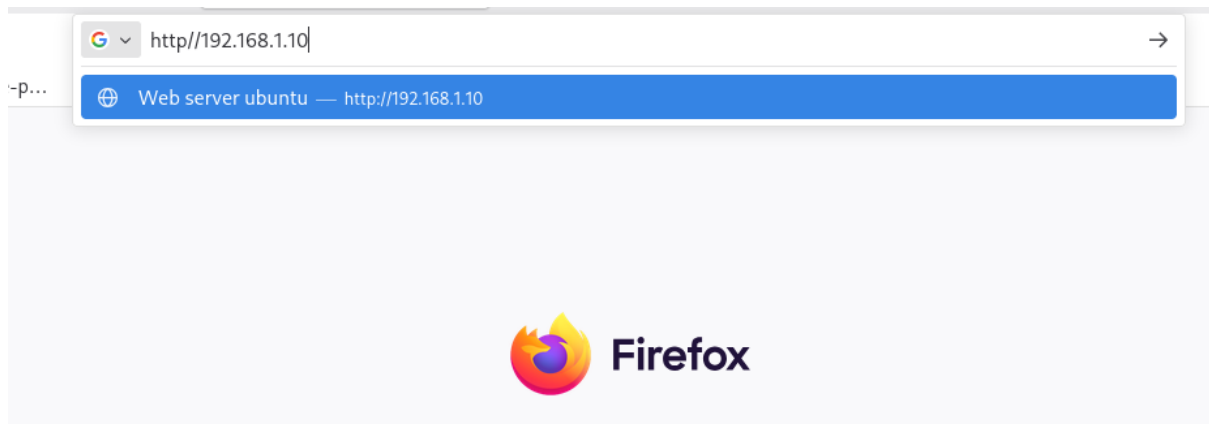
  

```

Frame 57: Packet, 631 bytes on wire (5048 bits), 631 bytes captured (5048 bits) on interface
Ethernet II, Src: VMware_f2:a9:2f (00:0c:29:f2:a9:2f), Dst: VMware_a9:73:00 (00:0c:29:a9:73:00)
Internet Protocol Version 4, Src: 192.168.1.10, Dst: 192.168.1.20
Transmission Control Protocol, Src Port: 80, Dst Port: 58440, Seq: 1, Ack: 354, Len: 565
Hypertext Transfer Protocol
Line-based text data: text/html (9 lines)
0000 00 0c 29 a9 73 00 00 0c 29 f2 a9 2f 08 00 45 00 ... ) s... ).../...E
0010 02 69 4a 33 40 00 40 06 6a ed c0 a8 01 0a c0 a8 ... i33@ @ j...
0020 01 14 00 50 e4 48 d4 18 b4 f1 6f 8b 02 1b 80 18 ... P.H... o...
0030 01 fb c1 0e 00 09 01 01 08 0a 3e 00 10 53 b2 cd ... > S...
0040 00 48 48 54 54 50 2f 31 2e 31 20 33 30 31 20 4d ... HTTP/1.1 301 M
0050 0f 76 65 64 20 50 65 72 6d 61 6e 65 6e 74 6c 79 ... oved Per manently
0060 0d 0a 41 61 74 65 3a 20 54 75 65 2c 20 31 37 20 ... Date: Tue, 17
0070 46 65 62 20 32 30 32 36 20 32 33 3a 31 37 3a 33 ... Feb 2020 23:17:3
0080 34 20 47 4d 54 6d 0a 53 65 72 76 65 72 3a 20 41 ... 4 GMT. S erver: A
0090 70 61 63 68 65 2f 32 2e 34 2e 36 34 20 28 55 62 ... pache/2. 4.64 (Ub
00a0 75 6e 74 75 29 6d 0a 4c 6f 63 61 74 69 6f 6e 3a ... untu) L ocation:
00b0 20 68 74 74 70 73 3a 2f 2f 31 39 32 2e 31 36 38 ... https:/ /192.168
00c0 2e 31 2e 31 30 2f 6d 0a 43 6f 6e 74 65 6e 74 2d ... .1.10/ Content-
00d0 4c 65 6e 67 74 68 3a 20 33 30 37 6d 0a 4b 65 65 ... Length: 307. Kee
00e0 70 2d 41 6c 69 76 65 3a 20 74 69 6d 65 6f 75 74 ... p-Alive: timeout
00f0 3d 35 2c 20 6d 61 78 3d 31 30 30 6d 0a 43 6f 6e ... =5, max= 100. Con
0100 6e 65 63 74 69 6f 6e 3a 20 4b 65 65 70 2d 41 6c ... nection: Keep-Al
0110 69 76 65 6d 0a 43 6f 6e 74 65 6e 74 2d 54 79 70 ... ive. Con tent-Typ
0120 65 3a 20 74 65 78 74 2f 68 74 6d 6c 3b 20 63 68 ... e: text/ html; ch
0130 61 72 73 65 74 3d 69 73 6f 2d 30 38 35 39 2d 31 ... arset=iso-8859-1

```

- **Résultat de la redirection** :



### 3.3. Défense Active (Fail2Ban)

- **Démarche :** Installation de Fail2Ban pour bannir toute IP échouant à 3 tentatives de connexion.

La protection active a été configurée dans un fichier local pour garantir sa persistance:

- **Fichier :** `/etc/fail2ban/jail.local`
- **Configuration du filtre SSH :**



```
[sshd]
enabled = true
port = 2222
filter = sshd
logpath = /var/log/auth.log
maxretry = 3
findtime = 10m
bantime = 1h
~
```

**Simulation :** La machine Debian échoue 3 fois l'authentification et est bloquée

```
prcx23@debian:~$ ssh prcx23@192.168.1.10 -p 2222
prcx23@192.168.1.10's password:
Permission denied, please try again.
prcx23@192.168.1.10's password:
Permission denied, please try again.
prcx23@192.168.1.10's password:
prcx23@192.168.1.10: Permission denied (publickey,password).
prcx23@debian:~$ ssh admin@192.168.1.10 -p 2222
ssh: connect to host 192.168.1.10 port 2222: Connection refused
prcx23@debian:~$
```

```
prcx23@ubuntu-server:~$ sudo fail2ban-client status sshd
Status for the jail: sshd
|- Filter
| |- Currently failed: 0
| |- Total failed: 3
| \- Journal matches: _SYSTEMD_UNIT=ssh.service + _COMM=sshd
- Actions
  |- Currently banned: 1
  |- Total banned: 1
  \- Banned IP list: 192.168.1.20
prcx23@ubuntu-server:~$ _
```

### III. Conclusion

Ce projet a permis de démontrer que la sécurité d'une infrastructure ne repose pas sur l'accumulation d'outils isolés, mais sur la **complémentarité de couches défensives**.

La mise en œuvre du changement de port (2222) et du chiffrement (HTTPS) a mis en évidence qu'une configuration est inefficace sans une phase de **validation par l'audit**.

L'utilisation de **Nmap** et **Wireshark** a ainsi été déterminante pour confirmer l'étanchéité des services.

L'expérience montre que si le pare-feu constitue la première ligne de défense périmétrique, l'intégration d'une solution de défense active telle que **Fail2Ban** est indispensable pour automatiser la réponse aux incidents en temps réel. En conclusion, ce projet souligne qu'une administration système rigoureuse exige une remise en question constante des configurations, laquelle doit impérativement être validée par des tests d'intrusion et une analyse de flux.

#### IV. Webographie

- **Documentation officielle Ubuntu (UFW) :**  
<https://help.ubuntu.com/community/UFW>
- **Apache HTTP Server Project (SSL/TLS):** <https://httpd.apache.org/docs/2.4/ssl/>
- **Fail2Ban Project Wiki:** <https://www.fail2ban.org/>
- **Guide de sécurisation SSH (OpenSSH) :**  
<https://www.ssh.com/academy/ssh/hardening>