Rapport de projet

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

Réalisé par : MABENGO Gloire Précieux

SOMMAIRE

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

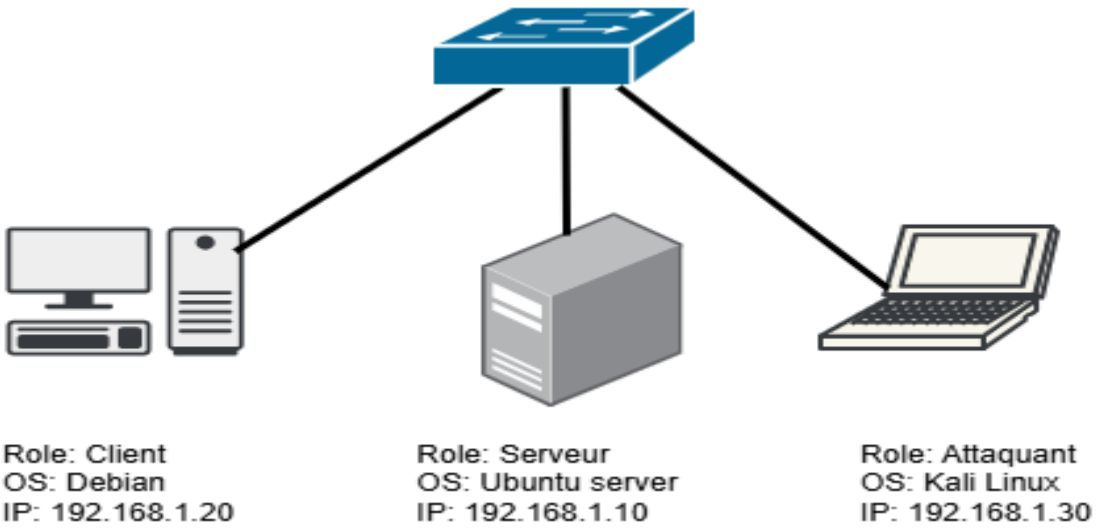
Le projet s'appuie sur une architecture virtualisée (VMWARE Workstation 25H2) comprenant trois entités :

- **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)**



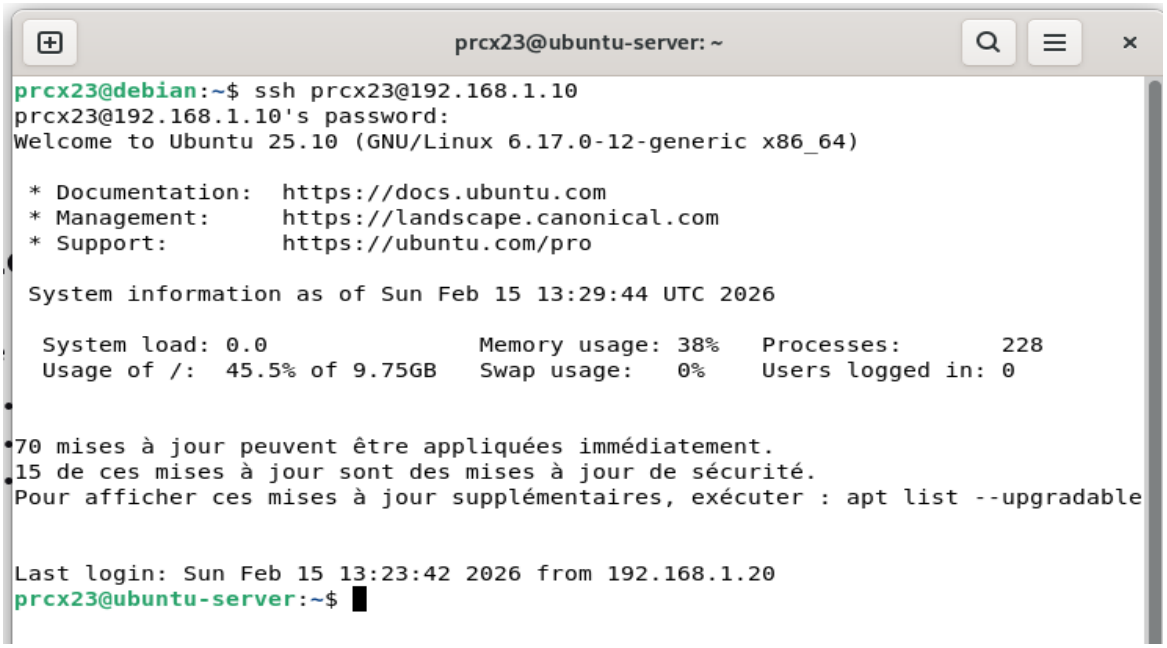
- **Fixation de l'adresse IP sur le serveur (modification du fichier** /etc/netplan/00-installer-config.yaml**)**

```
# This is the network config written by 'subiquity'
network:
  version: 2
  ethernets:
    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**

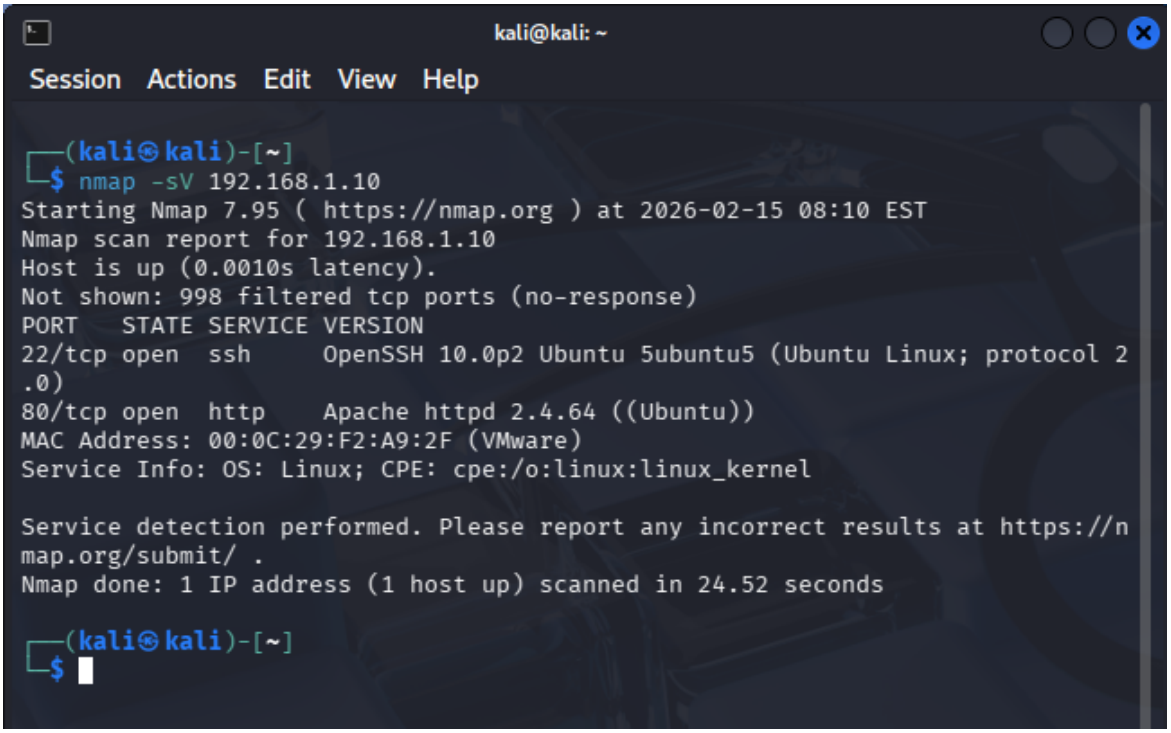


2. Scan de port

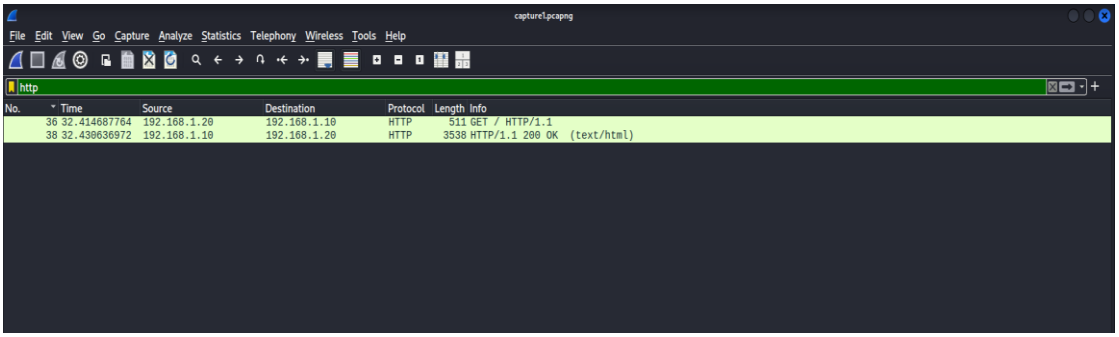
Un premier scan Nmap depuis Kali Linux a révélé la visibilité totale des services :

- **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.



- Vue du trafic HTTP avec wireshrark



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

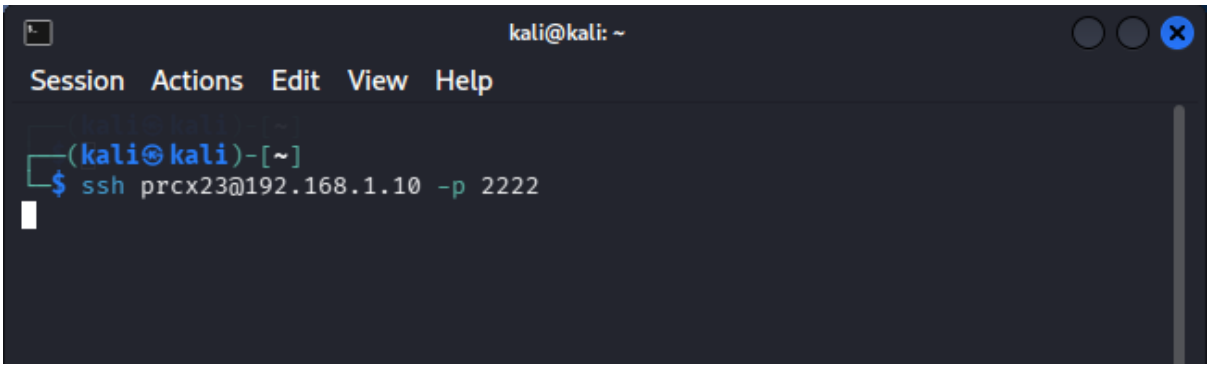
3.1. Sécurisation du SSH (Port 2222 et UFW)

- **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**).



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.

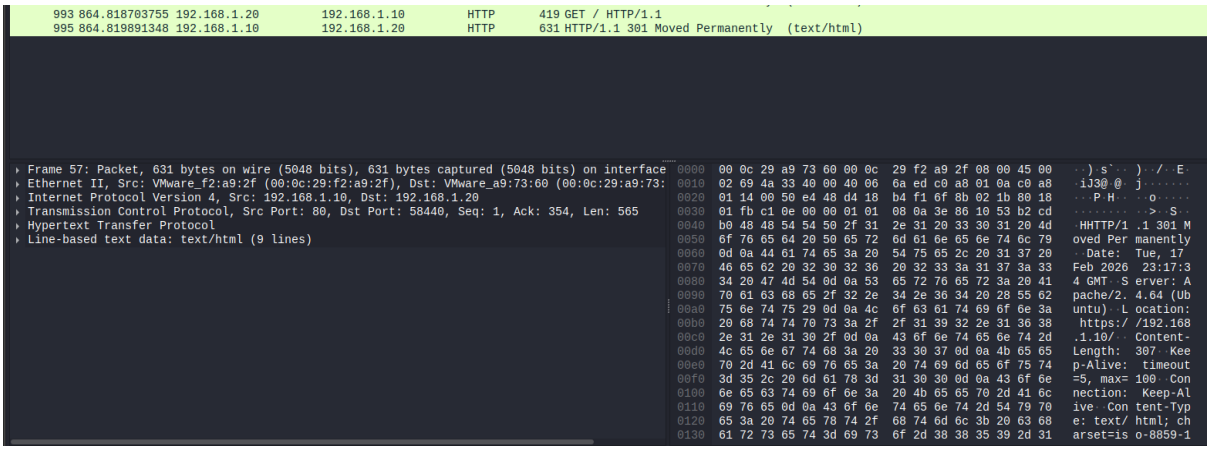
Nous avons utilisé le module rewrite d'Apache :

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

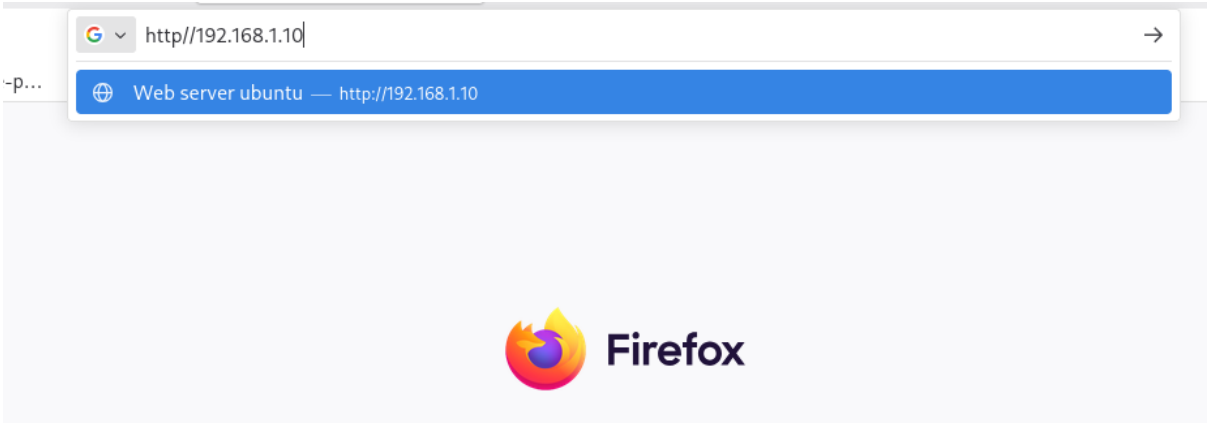
```
        #Rediriger le http vers https
        RewriteEngine On
        RewriteCond %{HTTPS} off
        RewriteRule ^(.*)$ https://%{HTTP_HOST}%{REQUEST_URI} [L,R=301]
</VirtualHost>
~
~
~
~
~
~
```

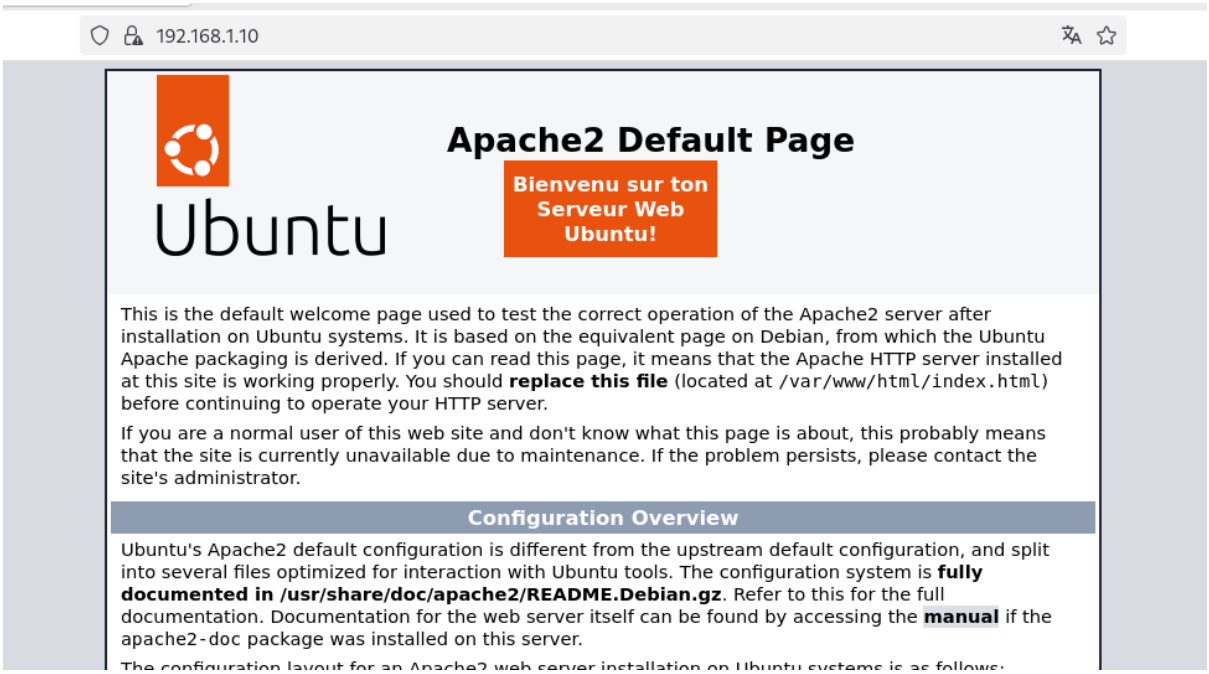
Directives ajoutées :

- **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).



- **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: ~
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 laboratoire 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